YAWL Editor 1.4.5 User Manual

Release 1.4.5

Document Control

Date	Author	Version	Change
2 Sep 2004	Sean Kneipp	0.1	Initial Draft
27 Sep 2004	Sean Kneipp	0.2	Data Perspective
17 Oct 2004	Sean Kneipp	0.3	Updated Editor Version
26 Oct 2004	Sean Kneipp	0.4	Updated Engine Integration
25 Nov 2004	Lindsay Bradford	0.5	Sync with Editor 1.1
04 Feb 2005	Lindsay Bradford	1	Finalised Release
25 Feb 2005	Lindsay Bradford	1.1-1	Sync with Editor 1.1-1
17 May 2005	Lindsay Bradford	1.2	Sync with Editor 1.2
27 Oct 2005	Lindsay Bradford	1.3	Sync with Editor 1.3
25 Feb 2006	Lindsay Bradford	1.4	Sync with Editor 1.4
19 May 2006	Lindsay Bradford	1.4.1	Sync with Editor 1.4.1
12 July 2006	Lindsay Bradford	1.4.2	Sync with Editor 1.4.2
10 Aug 2006	Lindsay Bradford	1.4.3	Sync with Editor 1.4.3
07 Dec 2006	Jessica Prestedge	1.4.5	Sync with Editor 1.4.5

Contents

Document Control ii
Welcome to YAWL 1
What is YAWL? 1
Obtaining the Latest Version of the YAWL Editor 1
The YAWL Project 2
YAWL Architecture 2
Getting Started 3
Copyright Notice 5
Launching the YAWL Editor 5
The YAWL Editor Workspace 6
Menu Toolbar 6
Palette Bar 8
Status Bar 9
Canvas 9
Menus Overview 9
Creating Your First Specification 11
Overview 11
The Scenario 11
Creating Your First Specification 11
Atomic Tasks 12
Decorating a Task 14
Composite Tasks 16
Multiple Atomic Tasks 17
Multiple Composite Tasks 20
Conditions 23
Changing the Starting Net 25
Changing the Appearance of Your Specification 26
Improving the Look of Flow Relations 26
Editing Objects 27
Changing Font Size 27
Advanced Specification Features 28
Cancellation Sets 28
Updating Data Type Definitions 30
Net Decomposition Detail / Updating Net Variables 32
Task Decomposition Detail / Updating Task Variables 34
Adding / Updating Task Parameters 36
Updating Flow Detail 41
End of Scenario 43
Set YAWL Engine Detail 43
Setting Multiple Instance Queries 45
Validating and Exporting 47
Illustrative Examples 50
Multiple Instance Task Example 50
Time Service Example 52
Known Issues 53
Troubleshooting 56
Acknowledgements 57

Welcome to YAWL

What is YAWL?

Based on a rigorous analysis of existing workflow management systems and workflow languages, we have developed a new workflow language: YAWL (Yet Another Workflow Language). To identify the differences between the various languages, we have collected a fairly complete set of workflow patterns. Based on these patterns we have evaluated several workflow products and detected considerable differences in expressive power. Languages based on Petri nets perform better when it comes to state-based workflow patterns. However, some patterns (e.g. involving multiple instances, complex synchronizations or non-local withdrawals) are not easy to map onto (high-level) Petri nets. This inspired us to develop a new language by taking Petri nets as a starting point and adding mechanisms to allow for a more direct and intuitive support of the workflow patterns identified.

YAWL can be considered a very powerful workflow language, built upon experiences with languages supported by contemporary workflow management systems. YAWL encompasses these existing workflow languages, and, in addition, has a formal semantics. Such an approach is in contrast with e.g. WfMC's XPDL which takes commonalities between various languages as a starting point and does not have formal semantics. YAWL provides support for the control-flow perspective, the data perspective and the resource perspective. The design environment also provides sophisticated analysis support.

Obtaining the Latest Version of the YAWL Editor

As new versions of the YAWL Editor are released to the public, they will be available for download at the YAWL website:

http://www.yawl-system.com/.

Developers interested in obtaining the YAWL Engine, can download the files from the YAWL website above. The source code for all of the YAWL components can be downloaded through the SourceForge Website:

http://sourceforge.net/projects/yawl.

The YAWL Project

For more information and progress on the YAWL project, visit the YAWL homepage:

http://www.yawl-system.com/.

YAWL Architecture

The following image, **Fig. 1.**, depicts the interaction between components of the YAWL Engine.

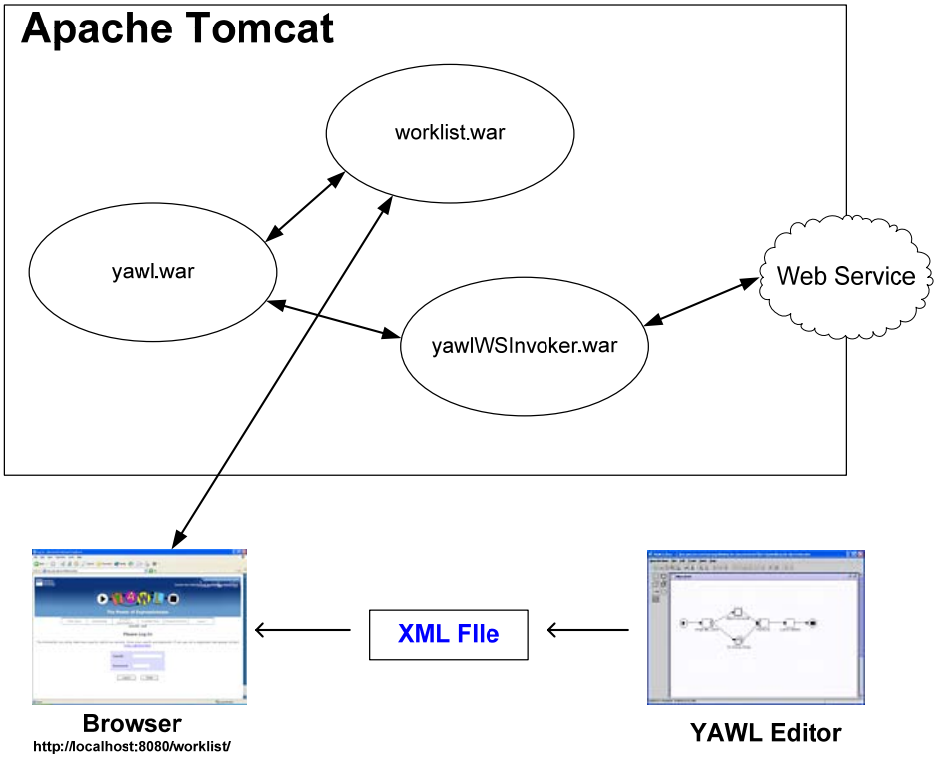


Fig. 1. The YAWL Components

Getting Started

What's New

Following is a list of new features and bug fixes introduced in this release and the previous release (Version 1.4.4) of the YAWLEditor:

- The Flow Relation button in the palette menu has been removed. Now to create a flow from one task to another all you need to do is left-click and drag from one connection point to the next valid connection.
- Upgraded the editor to use Engine 1.0-CR1. This includes switching the editor from using the engine war to the standalone engine jar for getting the engine library it needs. This is to better support the engine's installer including just the lite editor for distribution, but with full engine-enabled functionality.
- A side-effect of this is that the editor now requires Java 5 as its pre-requisite runtime JVM.
- Now, when a user selects a single task, the palette menu expands to allow the use to modify the split and join decorators for the task, without having to go to the popup menu for the task.
- Fixed a bug where a graph that was scaled would stop the user from being able to draw flows between elements.
- Now, on the label Element dialog, hitting enter in the label field will do the same as pressing the done button. This is a user convenience for quickly making label changes.
- The upgrade to 5.9.2 of JGraph broke certain cut and paste behaviour. This is now fixed to work with the latest JGraph.
- Altered how text is positioned with respect to the graphs upon export to PNG, JPG and print jobs. The text should now appear above the graph without writing over the top of it.
- Save files are now saved to a temporary file first and then copied to the final specified file name. If there is an editor crash whilst saving, only the temporary file will be corrupted.
- Fixed a bug where saving a specification with a task that has a single flow coming into or going out of it, then reloading the specification sees the user unable to add a decorator for that flow.
- Fixed a bug where right-clicking on the knee of a flow would see the pallet popup menu appear whenever the redrawn flow no longer lay underneath the mouse after knee removal.

- Upgraded the editor to fix some bugs in reset net analysis and to include reduction rules for specification analysis.

Copyright Notice

Copyright to this software and its source code is granted under the GNU Lesser General Public Licence (v2.1).

For detail on the permissions this licence grants you, please refer to http://www.gnu.org/copyleft/lesser.html.

This editor makes use of JGraph 5.9.2. This version of the JGraph library is covered under a n LGPL-style licence and is available separately from :

http://www.jgraph.com/.

This editor also makes use of code from the YAWL Engine and its support libraries. The engine and its support libraries are covered under their own individual copyright licences, detail of which may be found at the YAWL website:

http://www.yawl-system.com/.

In accordance with this editor's copyright licence, the source code may be obtained from the YAWL SourceForge website:

http://sourceforge.net/projects/yawl/.

Launching the YAWL Editor

1. Download the latest version from the YAWL SourceForge website:

   http://sourceforge.net/projects/yawl/.

2. Double click on the YAWLEditor.jar file to start the application and away you go.

The YAWL Editor Workspace

The first time you start the YAWL Editor, you will be presented with a blank canvas, with the instructions in the Status Bar asking you to open or create specification to begin.

Before you create your first specification, let us take a brief tour of the Editor's workspace and the elements within.

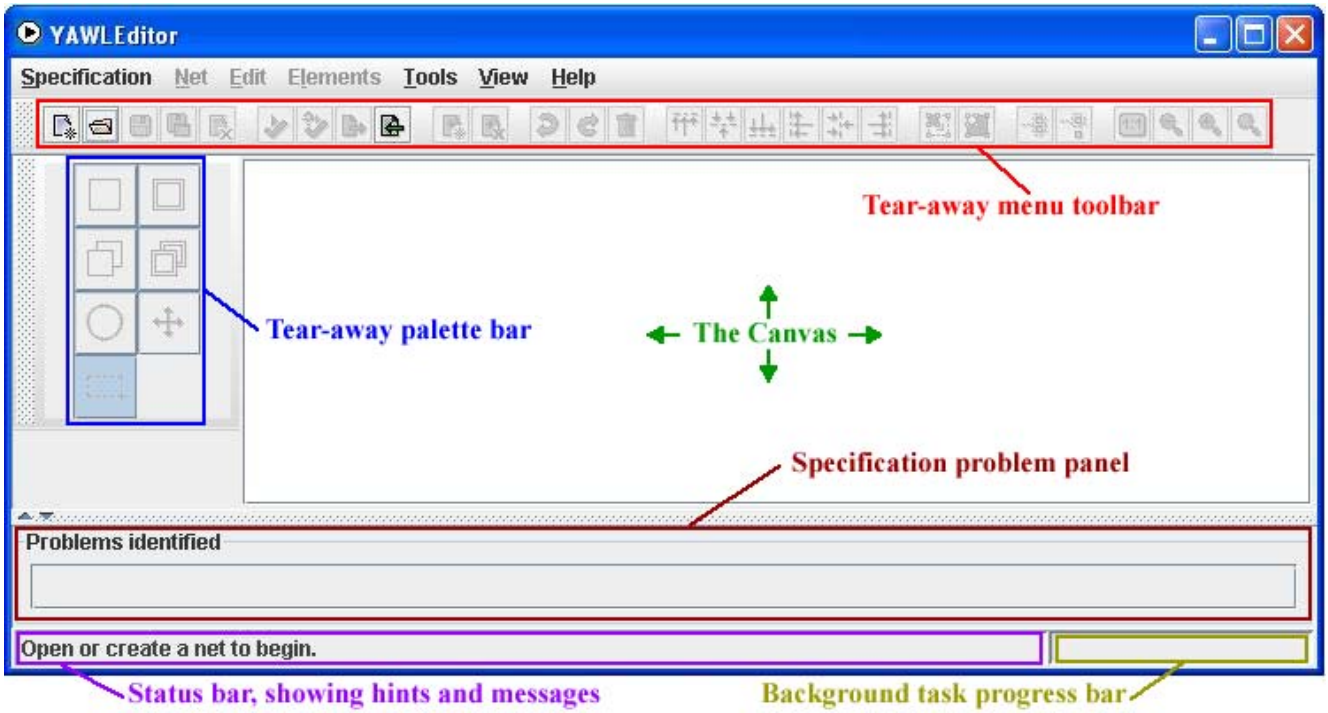


Fig. 2. The YAWLEditor Workspace

Menu Toolbar

The Menu Toolbar contains six groups of buttons to assist you in maintaining your YAWL specification. The menu can be repositioned by dragging the left-hand anchor bar.

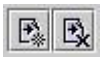
Specification Maintenance

This group of buttons provides you the standard options to create, save, open and close YAWL specification files.

Specification Verification, Analysis Exporting & Importing

These four buttons allow you to validate your specification again YAWL Engine workflow requirements, analyse your specification for deadlocks and other issues, export your workflow diagram into an XML file for uploading to the YAWL Engine, and import a YAWL Engine XML file into the editor for further alteration respectively.

Net Maintenance

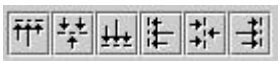


Your workflow diagrams are captured within Nets, of which a specification could contain many. You can use these buttons to create a new Net or remove existing Nets within your specification.

Edit Options

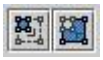
This group of buttons provides the standard Undo and Redo options as well as the option to delete the currently selected objects.

Alignment Options



These buttons can be used to assist with the alignment of objects within your specification, when multiple objects have been selected.

Object Sizes



To increase the size of an object within your specification, select the object(s) and then use these buttons.

Cancellation Sets

These buttons allow you to include in and/or exclude elements from the cancellation set of a task.

Zoom Options

These buttons allow you to apply zoom functionality to the currently selected net. In order, the buttons allow you to reset the zoom to the actual size, to zoom the entire net out, to zoom the entire net in, and to zoom into the currently selected net elements.

Palette Bar

The Palette Bar contains seven selector buttons that assist with creation, selection and positioning of objects within your specification. This menu can be repositioned by dragging the left-hand anchor bar.

The Palette bar is also accessible by right-clicking anywhere on a net that does not contain a net element.

Once an element is selected, it is possible to drop objects in the canvas by left-clicking the mouse button.

Atomic Task

Select this button to create an Atomic Task, which represent a single task to be performed by a human or external application.

Composite Task

Select this button to create a Composite Task, which is a container for another YAWL Net - with its own set of YAWL elements constrained by the same syntax.

Multiple Atomic Task

Select this button to create a Multiple Atomic Task, which allows you to run multiple instances of a task concurrently.

Multiple Composite Tasks

Select this button to create a Multiple Composite Task, which allows you to run multiple instances of a composite task concurrently.

Condition

Select this button to create a Condition, which is a way to represent state for the Net.

Marquee Selection

Select this button to activate the Marquee Selector, which will allow you to select individual or multiple objects by clicking and dragging your left mouse button. Please note: you can not create flows while using the Marquee Selector.

Drag Net Window

Select this button to drag the visible window of a net around that net.

Status Bar

The Status Bar provides useful hints throughout the creation of your specification and depending upon which object you are using in the Palette Bar.

Canvas

The Canvas is where you will be creating and editing your workflow diagram(s).

Specification Problem Table

The Specification problem table is where you see what problems are currently outstanding in the specification you are building.

Background task progress bar

The Background task progress bar shows work in progress for certain background tasks, like the saving of specification files.

Menus Overview

This section provides a brief overview of the YAWL Menus located along the top of the YAWL Editor.

Specification

The Specification Menu provides all the standard file options of Create, Open, Save, Close and Exit. It also contains options to validate (engine syntax), analyse (workflow semantics), export the specification to the YAWL Engine (XML format), import specifications from the YAWL Engine XML format, updating the specification's data type definition (in XMLSchema format), setting a specification's properties and for printing the specification.

Net

The Net Menu provides options to create, remove, set the starting net, specify decomposition detail, and resize the currently selected net. It also provides options to export a net to a PNG image file and for directly printing out the net.

Edit

The Edit Menu provides the standard options of Undo, Redo, Cut, Copy, Paste and Delete objects within your specification.

Elements

The Elements Menu allows you to align net elements within your specification, modify their size or add or remove net elements from task cancellation sets.

Tools

The Tools Menu allows you to specify a running engine that the editor should connect to so you can tailor your workflow designs to specific modules and web services installed in the engine. Included in this menu is an option for configuring specification analysis. With this release of the editor, an amount of default specification analysis is offered as part of the editor. If the **wofyawl** analysis utility is also supplied in the same directory as the editor, the configuration dialog will allow specification designers to configure and use wofwawl for additional specification analysis. Also, for resource allocation data, this menu offers a dialog to identify an organisation database from which resourcing information can be retrieved.

View

You can use this menu to turn off or on the Tooltips, which provide useful hints when your mouse is positioned over a toolbar button or an option in a pop-up dialog window, obtained by right-clicking on the canvas. It also allows you to change the font size used for element labels, toggle whether diagrams should be drawn anti-aliased and toggle whether grids should be drawn with diagrams. Finally, it allows users to iconify all nets, or deiconify all nets at once.

Help

The Help Menu provides details of the YAWL Editor copyright as well as Acknowledgements for all the dedicated YAWL contributors. Included is an "About the Editor" dialog which describes which components were used in the

editor's construction, compatibility issues, and a list of contributors who've added source code to the editor.

Creating Your First Specification

Overview

This next chapter will lead you through the process of creating a YAWL specification from beginning to end, through a series of brief lessons following a scenario.

You can either follow all the instructions including the scenario provided, from beginning to end, or skip straight to the section that you are interested in and follow the instructions.

Look for the **blue S** next to the instructions for specific details of the scenario.

The Scenario

The scenario that we will be following throughout this manual is the workflow of a student who has just completed their secondary study and is now looking to start their career.

This scenario will follow the path of a student who either enrols in a University to complete their tertiary eduction, or undertakes private study which will eventually lead them to getting a job and starting their new career.

Creating Your First Specification

1. Click on the Create a New Specification button, , at the top left of the Menu Toolbar, or click on **Specification** in the Menu and choose **Create Specification.**

   This will bring up a blank Net called "New Net 1" which will be, by default, the starting net of the workflow. For details on selecting a starting net, consult the *Changing the Starting Net* section in this manual.

2. Alter the specification's properties as you feel appropriate. Click on **Specification** in the Menu and choose **Update Specification Properties.** A screen as per **Fig. 3** will appear.

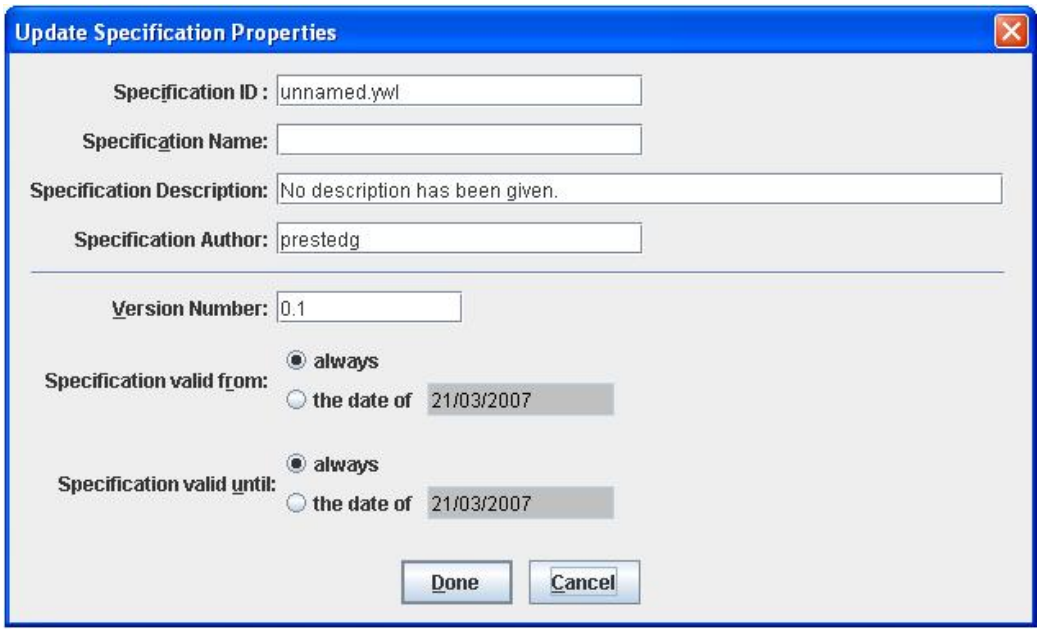


Fig. 3 Specification Properties Dialog

3. Rename this Net by clicking on the **Net** Menu and choosing **Update Net Detail**.

4. Enter the new name of the Net in the "Decomposition Label" field, then click the **Done** button.

   Decomposition Variables will be explained later in the "Net Decomposition Detail" section of this manual.

Change the name of the Net in the Decomposition Label, to "My Career". This Net will be the primary net for our scenario.

5. You are now ready to start drawing your specification.

Atomic Tasks

1. Click on the "Add an Atomic Task" button, , in the Palette Bar, or right click in the whitespace of the Net and choose **Atomic Task**.

2. Position your mouse just to the right of the Input Condition (the symbol), and click the left mouse button once to place an Atomic Task

3. Set the decomposition of this task by right clicking on the Atomic Task and choosing **Select Task Decomposition**.

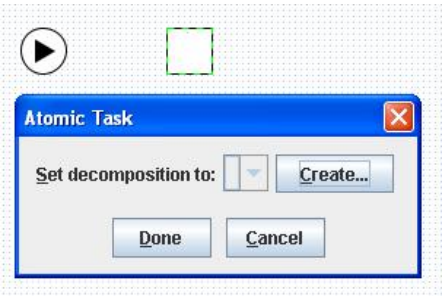


Fig. 4. The "Update Task Decomposition " dialog

4. Press the **Create…** button, and in the following window, "Update Task Decomposition", enter the decomposition's label.

   See the "Select Task Decomposition" section of this manual for a full explanation of its features.

Set the label to "Begin My Career", and click the **Done** button.

Note that by default, a task takes on the label of the decomposition that it is tied to (several tasks are allowed to share the same decomposition). Once you've task created your task, you are free to relabel the task to whatever you like. This can be done by right-clicking on the task and choosing **Set Label…** from the pop-up menu. This will not change the name of the decomposition to which the task is tied.

5. Connect the Input Condition to your Atomic Task by finding the flow connectors, indicated by two parallel blue lines on the sides of the objects, and hold the left mouse button down and draw a line from the flow connector on the Input Condition to the one on the Atomic Task. The editor will indicate which connection points are valid by drawing a blue box around suitable connectors as your mouse passes over them.

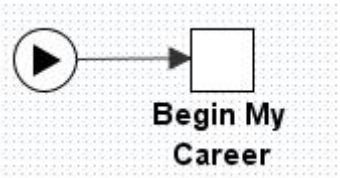


Fig. 5. An established Flow Relation

That's it! Your Atomic Task is set.

6. Repeat the process for the following Atomic Tasks in order:
   - Go to University
   - Get A Job
   - Begin My Career
7. Link the Begin My Career task to the Output Condition (the ◉ symbol), as per **Fig. 6**.
8. Finally check the validity of specification by clicking on the Validate this Specification button, , in the Menu Toolbar or click on **Specification** in

the Menu and choose **Validate Specification**. If all things are going to plan, then you should receive a confirmation saying that there were no errors detected.

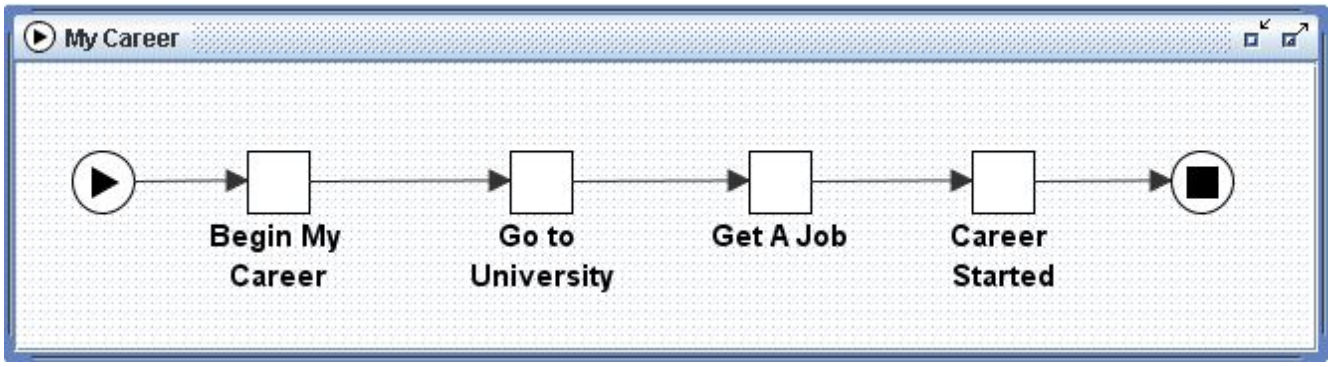


Fig. 6. The "My Career" Net

Decorating a Task

Decorating a task is the process of adding splits or join conditions on task.

By putting a split on a task, you are telling the task that when it has been completed, it's succeeding task could be one or more tasks. Here are the possible splits for a task:

- No split (no split on task)
- AND split
- OR split
- XOR split

By putting a join on a task, you are telling the task that it could become available through the completion of one or more preceding tasks. Below are some possible joins for a task:

- No join (no join on task)
- AND join
- OR join
- XOR join

For more information on join and split types, please consult the YAWL technical papers on the YAWL website.

Creating Splits and Joins

To create a split or join:

1. Ensure that your palette is in "Net Element Selection" mode.

   Select a task. When a single task is selected the palette menu will expand

(see **Fig. 7.**) to include two tabs that allow you to decorate a task with a split and/or join.

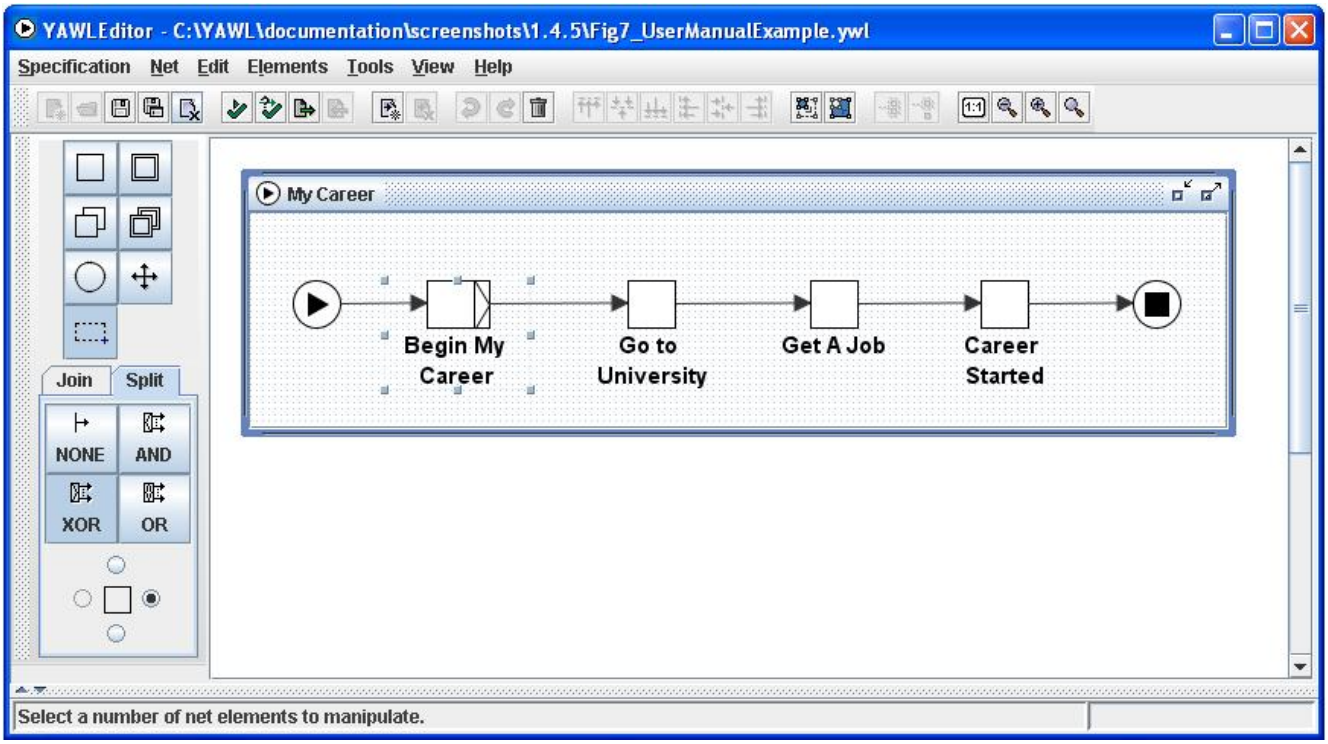


Fig. 7. Decorating a Task

2. Choose the required split or join and the orientation for the split or join to appear and then click **Done**.

Select your "Begin My Career" task and change the split to **Xor Split**. Then set the orientation to eastern edge of the task , as per **Fig. 7.**

Create a new Atomic task called "Do Private Study". This task will represent those students that choose not to go to University.

Finally, select your "Get A Job" task and decorate it with an **Xor Join**. Then set the orientation of the join to the western edge of this task.

3. Split and Join decorators allow you to connect several Flow Relations from and to your task respectively.

Create a flow relation from "Begin My Career" to "Do Private Study", then create another flow relation from "Do Private Study" to "Get A Job", as per **Fig. 8.**

4. Don't forget to check the validity of your specification.

Hint: If you are having troubles with positioning your tasks, the alignment tools are a big help.

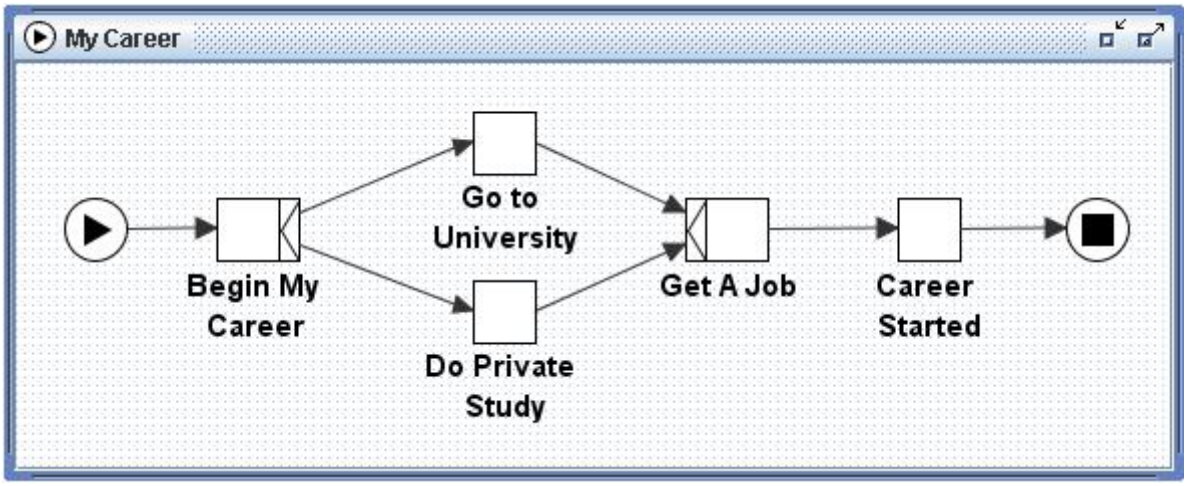


Fig. 8. XOR Split and Join

When "Begin My Career" has been completed, a choice must be made on which of the two tasks ("Go To University" or "Do Private Study") will be followed (XOR Split). "Get a Job" will become available after the completion of the task selected at the point of the XOR split.

Composite Tasks

Composite tasks are place holders for other YAWL Nets. That is, you can create another workflow in a separate Net, which is represented by the composite task in other nets.

To create a Composite Task:

1. Click on the Composite Task button, , in the Palette Bar or right click in the whitespace of the Net and choose **Composite Task**.

We are going to replace our existing "Go to University" Atomic Task, with a composite task, so choose the Enter Marquee Selection Mode button, , on the Palette Bar, click on the "Go to University Atomic Task" and press the Delete key on the keyboard. We will add in the new composite task next.

2. Place your Composite Task in your Net.

Reconnect the Flow Relations from "Begin My Career" to the new Composite Task, and from new Composite Task to "Get a Job".

3. Create a new Net by clicking on the Create a new Net button, , on the Menu Toolbar, or click on **Net** in the Menu and choose **Create Net**.

4. Choose a name for this Net by clicking on the **Net** Menu and choosing **Update Net Detail**.

We are going to call this new Net "Attend University".

5. Return to your original Net and right click on your Composite Task and choose **Unfold to net…**. You will then be given a drop-down list with all the Nets available – choose the Net this task is to represent and then click **Done**.

Choose “Attend University”.

6. You can now fill out the detail of your new ``Attend University'' Net.

Create the following Atomic Tasks in order and then link them with Flow Relations as per .,.,., and don’t forget to check for validity:

- Enrol
- Do Subjects
- Pass All Subjects
- Get Degree

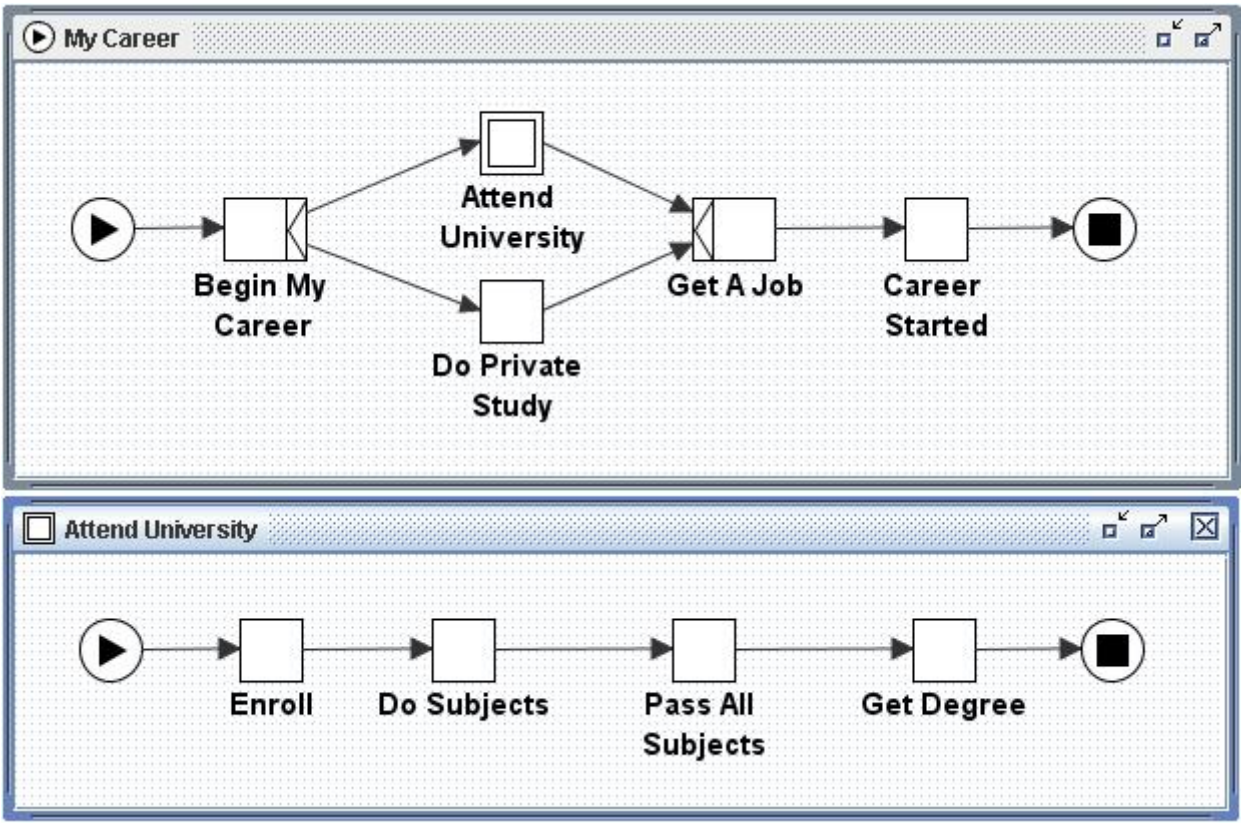


Fig. 9. "Attend University" Composite Task and its matching Net

Multiple Atomic Tasks

Multiple Atomic Tasks allow you to run multiple instances of a task concurrently.

To create a Multiple Atomic Task:

1. Click on the Add Multiple Atomic Task button, , in the Palette Bar or right click in the whitespace of the Net and choose **Multiple Atomic Task**.

Go back to the "My Career" Net. We are going to replace our existing "Do Private Study" Atomic Task, with a Multiple Atomic task, so choose the Enter Marquee Selection Mode button, , on the Palette Bar, click on the "Do Private Study" Atomic Task and press the Delete key on the keyboard. We will add in the new Multiple Atomic task next.

2. Place your Multiple Atomic Task in your Net and set the name of this task by right clicking on the task and choosing **Select Task Decomposition**.

Call this task the same name as before by selecting the "Do Private Study" decomposition from the drop-down list.

Reconnect the Flow Relations from "Begin My Career" to "Do Private Study", and from "Do Private Study" to "Get A Job", as per **Fig. 10.**

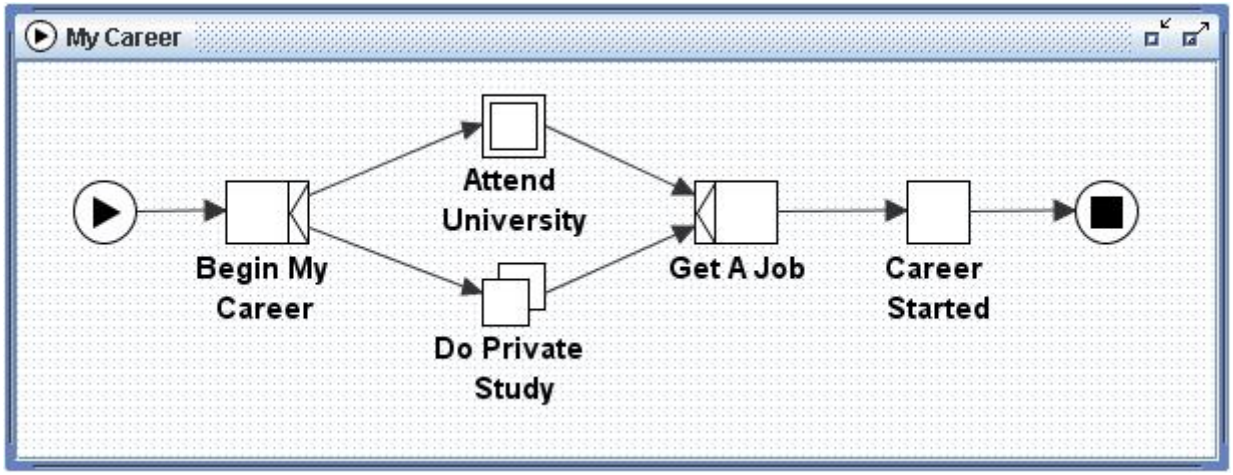


Fig. 10. Adding a Multiple Atomic Task

3. You will now need to set the parameters of the Multiple Atomic Task. Right click on the task and choose **Set Instance Detail**. Ensure that you are viewing the "Bounds" tab of the dialog, as per **Fig. 11.**

4. Set the Minimum Instances value. This is the minimum number of instances of this task that will be started when the task needs processing.

Set the Minimum Instances to 5.

5. Set the Maximum Instances value. This is the maximum number of instances of this task that can be created.

Set the Maximum Instances to 100

6. Set the Continuation Threshold value. The moment all instances created have completed or as many instances as the Continuation Threshold specified have completed, the multiple instance task itself is considered

completed, which triggers relevant outgoing flow relations from this task.

Set the Continuation Threshold to 50.

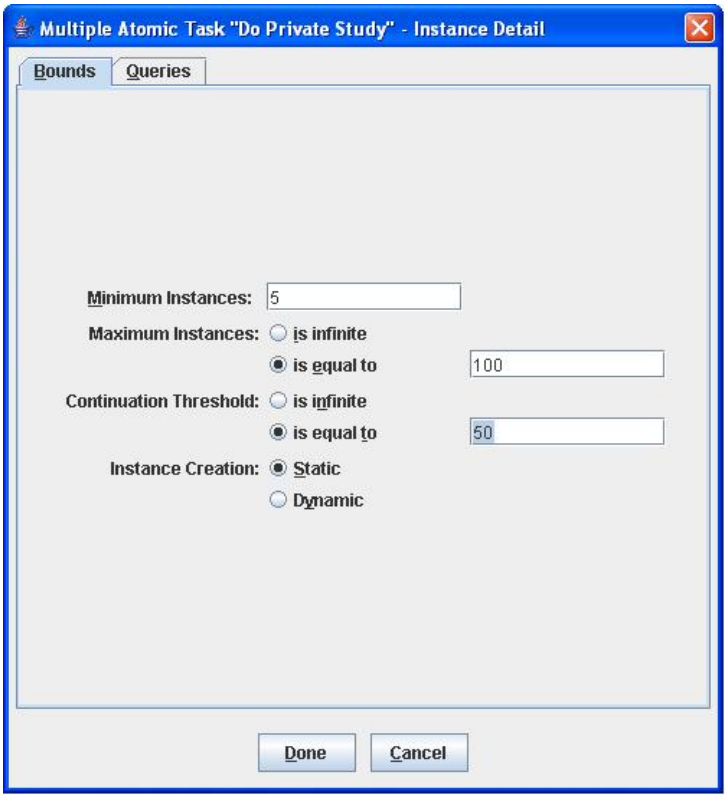


Fig. 11. Instance Bounds on Multiple Instance Tasks

7. Choose the Instance Creation. If a multi-instance task has static creation mode, when the task is started the number of instances created is decided then, and cannot be altered during the execution of the task. Conversely, if a task has dynamic creation mode then new instances of the task can start once the initial minimum instance number of tasks has started, but before the maximum number has been received.

   Set the Instance Creation type to “Static”.

8. Click **Done**.

With the values set in the scenario, the Do Private Study task is indicating that a maximum of 100 instances of this task can be executed before triggering the task’s outgoing flow. , A minimum of five instances will be created, and once 50 instances have completed, the outgoing flow relations to ‘’Get A Job” will trigger.

The “Queries” tab of the Multiple Instance dialog will be explained later in the “Setting Multiple Instance Queries” section of this manual.

Multiple Composite Tasks

Multiple Composite Tasks allow you to run multiple instances of the Net represented by a multiple composite task, concurrently.

To create a Multiple Composite Task:

1. Click on the Add Multiple Composite Task button, , in the Palette Bar or right click in the whitespace of the Net and choose **Multiple Composite Task**.

Go to the "My Career" Net. We are going to replace our existing "Do Private Study" Multiple Instance Task, with a Multiple Composite task, so choose the Enter Marquee Selection Mode button, , on the Palette Bar, click on the "Do Private Study" Task and press the Delete key on the keyboard. We will add in the new Multiple Composite task next.

2. Place your Multiple Composite Task in your Net.

   Reconnect the Flow Relations from "Begin My Career" to the new Multiple Composite Task, and from the new Multiple Composite Task to "Get a Job".

3. You will now need to set the parameters of the Multiple Composite Task. Right click on the task and choose **Set Instance Detail**.

4. Set the Minimum Instances value. This is the minimum number of instances of this task that will be started when the task needs processing.

Set the Minimum Instances to 5.

5. Set the Maximum Instances value. This is the maximum number of instances of this task that can be created.

Set the Maximum Instances to 100

6. Set the Continuation Threshold value. The moment all instances created have completed or as many instances as the Continuation Threshold specified have completed, the multiple instance task itself is considered completed, which triggers relevant outgoing flow relations from this task.

Set the Continuation Threshold to 50.

7. Choose the Instance Creation. If a multi-instance task has static creation mode, when the task is started the number of instances created is decided then, and cannot be altered during the execution of the task. Conversely, if a task has dynamic creation mode then new instances of the task can start once the initial minimum instance number of tasks has started, but before the maximum number has been received.

Set the Instance Creation type to "Static".

8. Click **Done**.

9. Create a new Net by clicking on the Create a new Net button, , on the Menu Toolbar, or click on **Net** in the Menu and choose **Create Net**.

10. Give the new Net a name by clicking on the **Net** Menu and choosing **Update Net Detail.**

We are going to call this new Net “Study Privately”.

11. Return to your original Net and right click on your Multiple Composite Task and choose **Unfold to Net**. You will then be given a drop-down list with all the Nets Available – choose the Net for this task to initiate and then click **Done**.

Choose “Study Privately”:

12. You can now complete your new “Study Privately” Net represented by your Composite Task.

Create the following Atomic Tasks in order and then link them with Flow Relations as per **Fig. 12.**

- Read a Book
- Feel Smarter

Don’t forget to validate your specification.

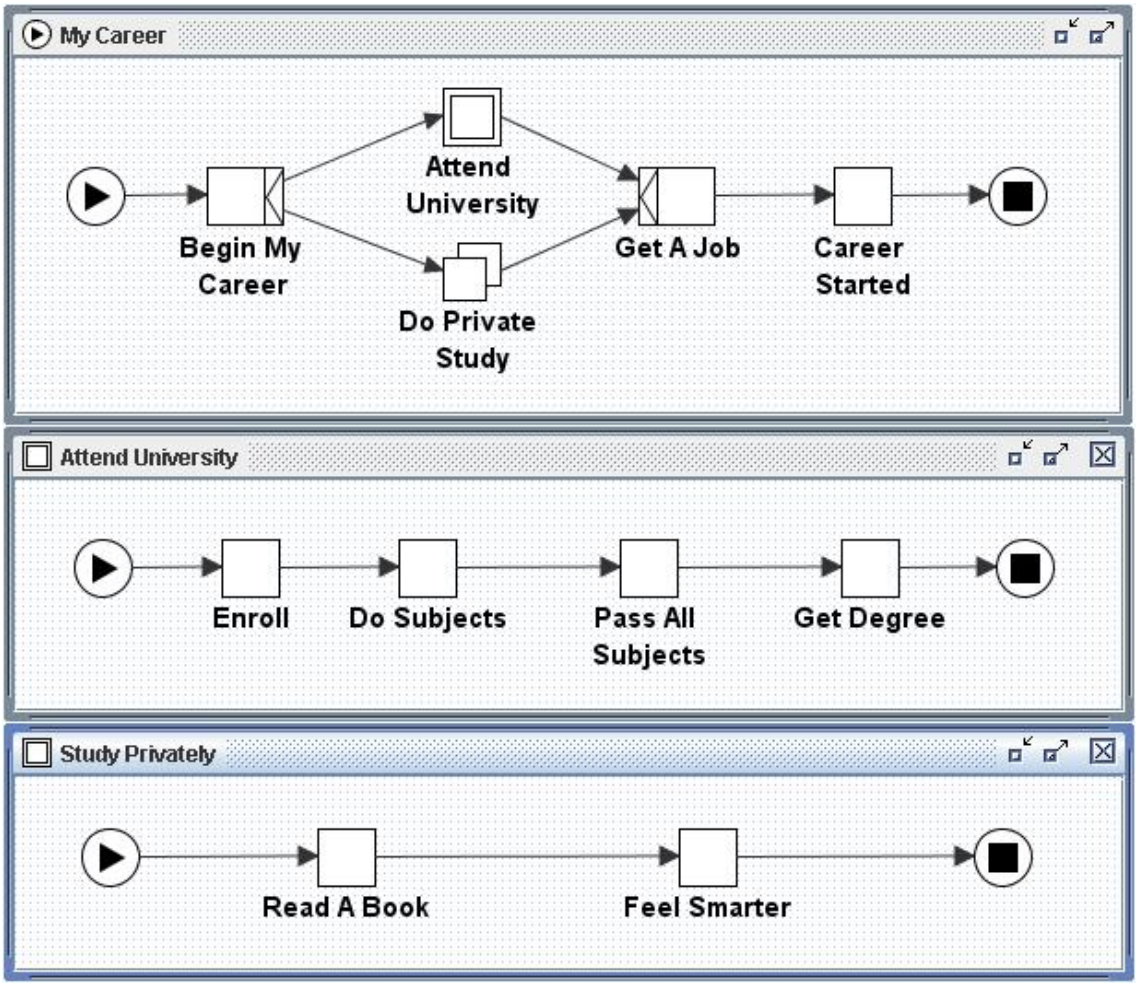


Fig. 12. "Privately Study" Multiple Composite Task and its Net

Conditions

Conditions are ways to represent a state, in between tasks, for the workflow.

To create a Condition:

1. Click on the Add a Condition button, , in the Palette Bar or right click in the whitespace of the Net and choose **Condition**.

Go to the "Study Privately" Net. We are going to place a loop Condition after the Read a Book Atomic Task, to determine whether we gained any knowledge from the book. We will add the new Condition next.

2. Place your Condition in your Net and set the name by right clicking on the Condition and choosing **Set Label**.

Call this Condition "Knowledge Gained?"

3. Now link to the Condition using the Flow Relations object.

Select the Enter Marquee Selection Mode button, , from the Palette Bar and click on the Flow Relation going from the Read a Book Atomic Task to the Feel Smarter Atomic Task and press the Delete button on the keyboard.

Create a Flow Relation from "Read A Book" to "Knowledge Gained?"

4. Create a Flow Relation from your Condition to a task.

Set the Flow Relation from "Knowledge Gained?" Condition to "Feel Smarter" Atomic Task.

5. Create another Flow Relation from your Condition to another task to signify the two possible flows from the Condition.

Before we create our second flow relation from our Condition, create another Atomic Task and call it "Look for Another Book".

Change the Join Decoration for Atomic Task "Read a Book" to an OR Join, with the orientation being West, by right clicking on Read a Book and choosing **Decorate**. Click **Done**.

Finally create the Flow Relation from the "Knowledge Gained?" Condition back to the OR Join of the "Read A Book" Atomic Task, as per **Fig. 13.**

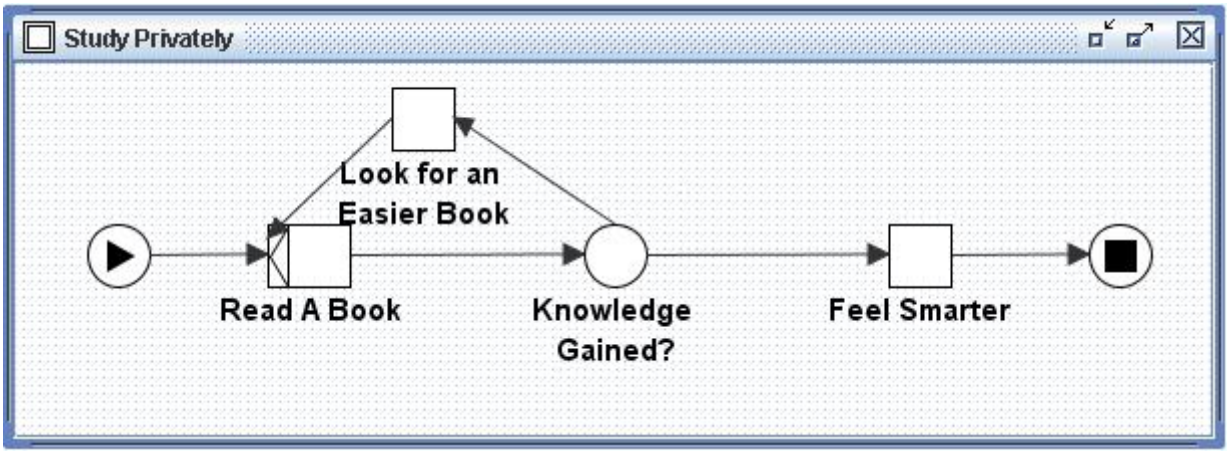


Fig. 13. Condition Loop for "Study Privately" Net

6. Validate your specification. Validation should fail and report errors as per **Fig. 14.** The problem here is that the "Study Privately" multiple instance composite task needs to have more information specified for it to be valid.

   For setting data detail of multiple-instance tasks, please see the "Setting Multiple Instance Queries" section of this manual.

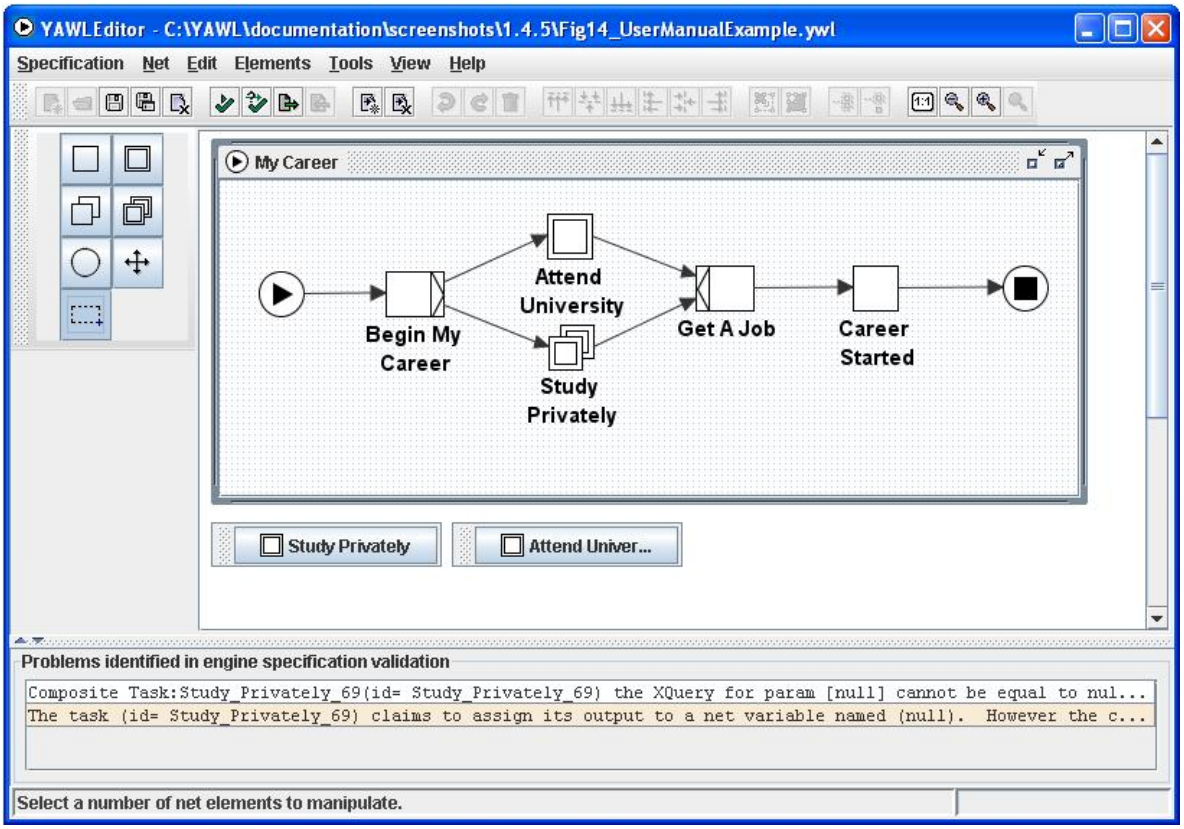

YAWLEditor - C:\YAWL\documentation\screenshots\1.4.5\Fig14_UserManualExample.ywl
Specification Net Edit Elements Tools View Help
My Career
Begin My Career
Attend University
Study Privately
Get A Job
Career Started
Study Privately
Attend Univer...
Problems identified in engine specification validation
Composite Task:Study_Privately_69(id= Study_Privately_69) the XQuery for param [null] cannot be equal to nul...
The task (id= Study_Privately_69) claims to assign its output to a net variable named (null). However the c...
Select a number of net elements to manipulate.

Fig. 14 - Validation with unfinished Multiple-Instance Tasks

7. Remove the ``Study Privately'' multiple-instance composite task and replace it with an atomic composite task using the same decomposition, and re-drawing flows from "Begin My Career" and to "Get A Job" tasks. Your updated net should look like **Fig. 15**.

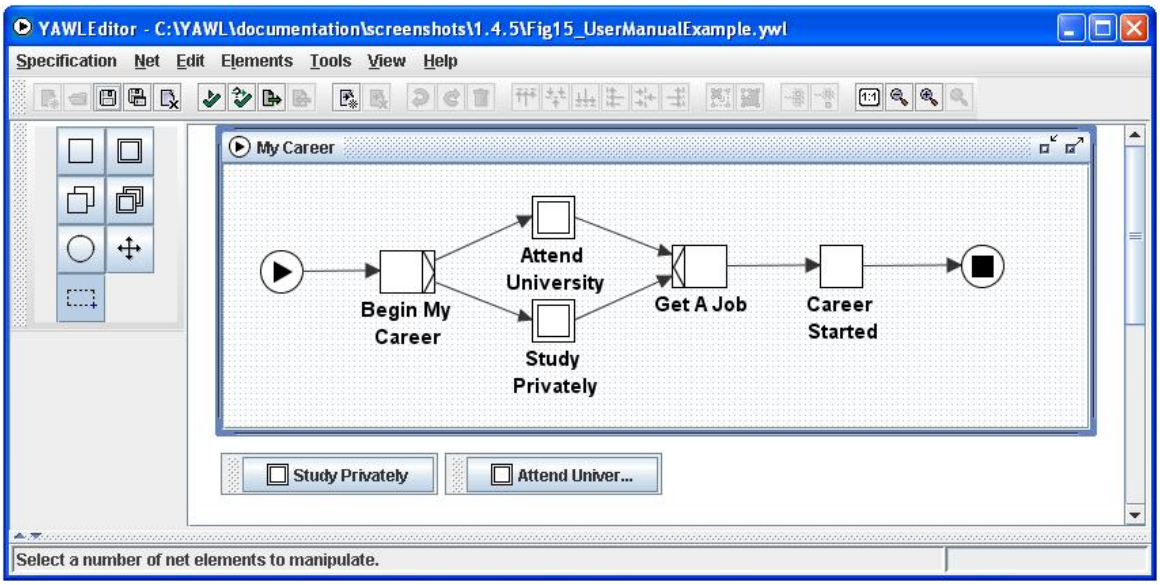


Fig. 15 - Making the "Study Privately " task an atomic composite

Changing the Starting Net

At any stage you can change the starting Net of the specification. To change the starting Net:

1. Select **Net** from the Menu.
2. Choose **Set Starting Net**.
3. From the Choose Starting Net window, click on the drop-down list and select a new starting Net.
4. Click **Done**.

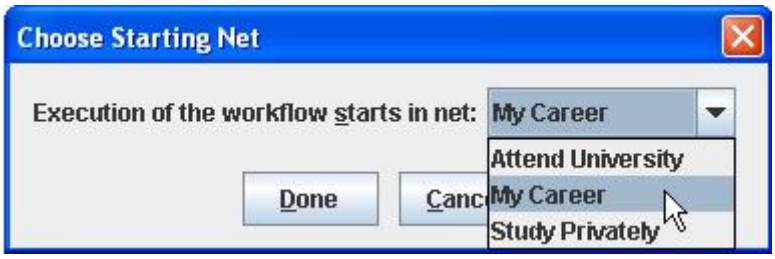


Fig. 16. Changing the Starting Net

Note that the starting net has an input condition symbol, ▶, in its title frame, and its minimised icon. All sub-nets have a composite task symbol, ▣, in their title frames and minimised icons.

Changing the Appearance of Your Specification

Improving the Look of Flow Relations

You can control and improve the look of the Flow Relations by adding in "knees".

1. Select the Enter Marquee Selection Mode button, , from the Palette Bar.

   Go to the "Study Privately" Net.

2. Right click on the position in the Flow Relation where you want to add a "knee", signified by a small square, , in the Flow Relation.

   Right click somewhere on the Flow Relation going from "Knowledge Gained?" Condition to "Look for an Easier Book" Atomic Task. Then left click on the knee created and drag it out to a more desirable location. You can add as many knees to a Flow Relation as you like.

3. Repeat the process for the Flow Relation between "Look For an Easier Book" and "Read A Book" tasks (see **Fig. 17.**).

You can now reconnect flow relations to other elements of a net, or different points on the same element. When the pallet button "Net Element Selection Mode", , is selected, left-click on a flow and drag it to another element. If a connection is possible, it will be highlighted, as described earlier. Release the mouse button to attach the flow to its new home.

4. Take the current flow relation, and move it from the top of the task to its side, as depicted in **Fig. 17**.

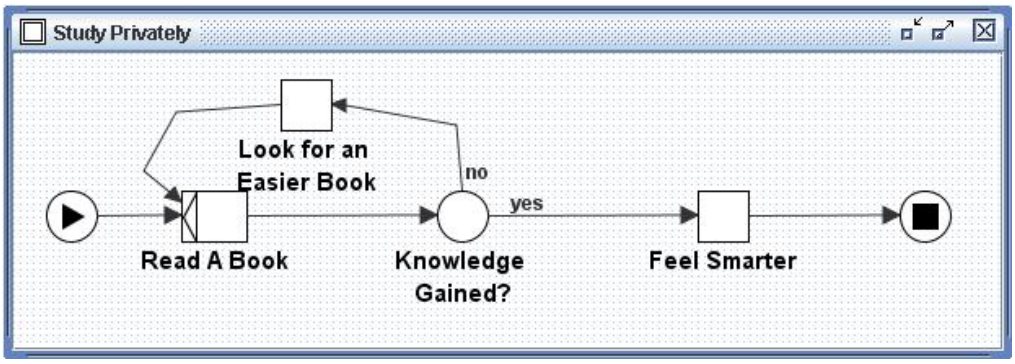


Fig. 17. Adding "knees" to a Flow Relation

It is also possible to add labels to flows. To do so, double click on a flow when the pallet button "Net Element Selection Mode", , is selected. A small editor will appear over the flow. Type your desired text, and commit the flow label by pressing the ENTER key. You may then drag that flow label around as desired.

5. Take the two flow relations that have recently had knees added to them. Attach the label “yes” to the flow relation going from the “Knowledge Gained?” condition to the “Feel Smarter” atomic task. Attach the label “no” to the flow relation going from the “Knowledge Gained?” condition to the “Look for an Easier Book” atomic task. Drag the labels about to a desired position, much like what has been done in **Fig. 17.**

Editing Objects

You can edit more than one object at a time by using the Enter Marquee Selection Mode tool. See **Fig. 18.**

1. Select the Enter Marquee Selection Mode button, , from the Palette Bar.
2. Click on the first object that you want to edit.
3. Hold down the shift key and then click on the other objects that you want to edit.
4. Now choose the **Edit** option from the Menu or continue holding down the shift key and right click on the mouse button. Below are the edit options:
   - Cut, Copy, Delete
   - Align
   - Size Increase / Decrease to change the appearance of the objects. This can also be done using the CTRL key plus Up or Down arrow on your keyboard.

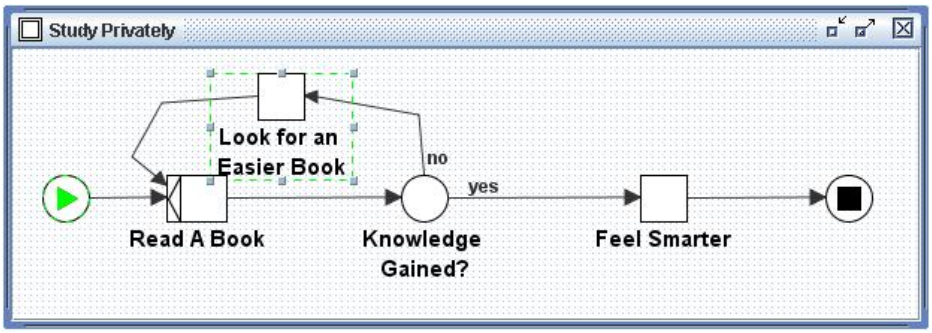


Fig. 18. Changing the Size of Multiple Objects

Note also, that whenever you have selected a number of net elements, pressing one of the arrow keys will move the selected elements in the direction of the arrow key, and pressing the CTRL key plus the A key will select all elements in the net with input focus.

Changing Font Size

You can change the size of the font used to label tasks and conditions.

1. Change the font size by clicking on the **View** Menu and choosing the **Label Font Size… option.**

Change the font size to 18.

Advanced Specification Features

Cancellation Sets

Cancellation Sets allow you to nominate any number of tasks, conditions or implicit conditions (which exist on flow relations between tasks, but are not drawn) for cancellation, upon the execution of a specified task. That is, once a specified task is executed in the workflow, all state within that task's cancellation set is removed.

To create a Cancellation Set:

1. First select the task that will initiate the Cancellation Set, using the Enter Marquee Selection Mode button, , from the Palette Bar.
2. Right-click on the task, then choose **View Cancellation Set**.

   The task will be coloured grey to indicate that this task is the cancellable task.

Go to the "My Career" Net. We are going to make the "Attend University" task a cancellable task to demonstrate in the workflow, that if a person executes this task, then they don't have time to perform the "Study Privately" task, therefore it should be removed as an option.

Right-click on "Attend University" and choose **View Cancellation Set** (as per **Fig. 19.**).

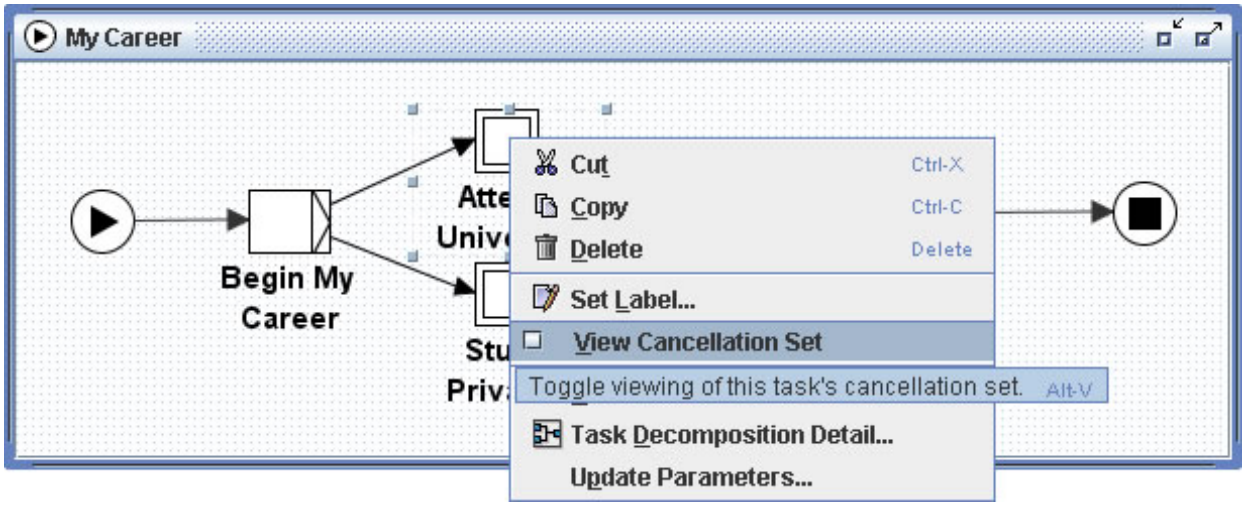


Fig. 19. Viewing a task's cancellation set

3. Next, using the Enter Marquee Selection Mode button, , from the Palette Bar, choose a task, condition or implicit condition to add to the Cancellation set. Hold down the shift key to select more than one object for cancellation.

Select the "Study Privately" task and the Flow Relation preceding it.

4. Click on the Add Selected Items to Visible Cancellation Set button, , on the Menu Toolbar.

   Items will be coloured red to indicate belonging to the cancellation set (see **Fig. 20.**).

Add the "Study Privately" task and the preceding Flow Relation to the cancellation set.

5. Once you have established the cancellation set, you can right-click on the cancellable task and tick-off **View Cancellation Set** option.

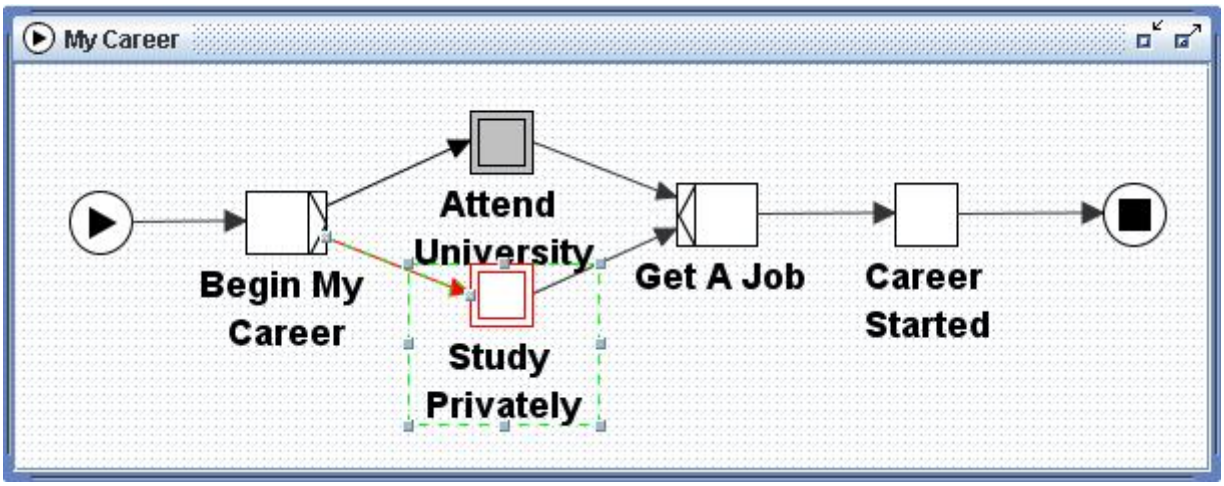


Fig. 20. Adding elements to a Cancellation Set

Notes about Cancellation Sets:

- A Cancellation Set that has been created will remain in the workflow, regardless of whether you have the **View Cancellation Set** option ticked.
- You can create multiple Cancellation Sets in your workflow, by selecting another task and choosing the **View Cancellation Set** option.
- All flows leading to or from conditions are not valid cancellation set members. Neither are the Input and Output conditions. The editor will ignore them if you select them for inclusion in a task's cancellation set.

To remove an element from a Cancellation Set:

1. First, make sure you have the **View Cancellation Set** option ticked.

   If it isn't ticked, select the task that is initiating the Cancellation Set, using the Enter Marquee Selection Mode tool from the Palette Bar and right-click, then choose **View Cancellation Set**.

2. Select the element for removal, using the Enter Marquee Selection Mode button, .

3. Click on the Remove Selected Items from Visible Cancellation Set button , ,on the Menu Toolbar.

Updating Data Type Definitions

Version 1.1 of the YAWL Editor introduces the ability for workflow designers to specify data that is processed by a workflow. XMLSchema is used to describe the valid data types that are available.

By default, a number of simple XMLSchema data types are supplied for variable definition, but if you need more complex data types, you can supply your own XMLSchema definition to describe them.

1. Select **Update Data Type Definitions** from the **Specification** Menu.

2. Enter your XMLSchema Data Type Definition into the dialog box. **Fig.28.**

3. If the definition text is green, your new data type is now available for defining Net or Task variables in your specification. If the text is red, there is something wrong with your data type definition, and the data type will not be available.

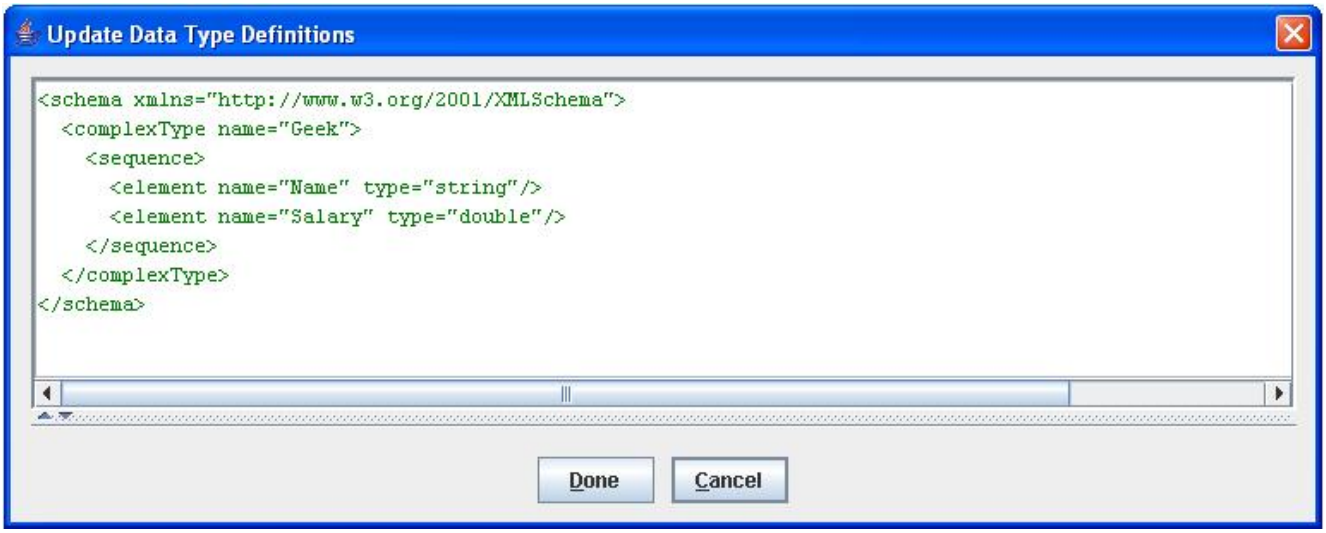


Fig. 21. Adding the "Geek" complex data type

When the text is red, the split-pane will reveal a table listing parse errors that were collected when determining the validity of the text supplied. An example of this is shown in **Fig. 22**.

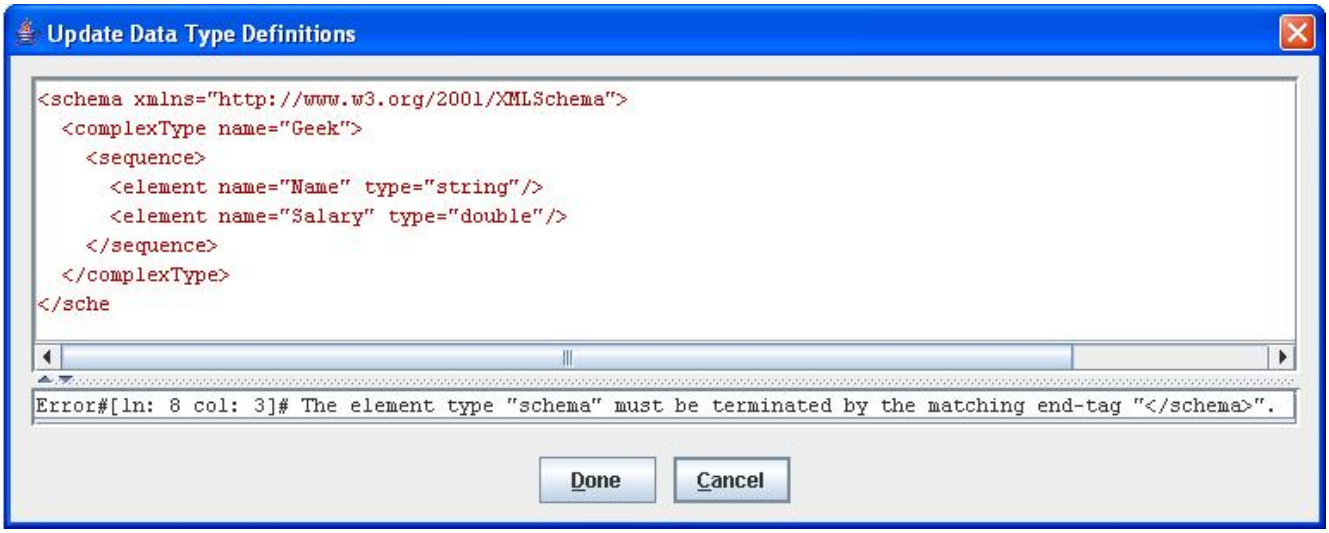


Fig. 22. When the "Geek" complex data type schema is invalid

Open the Data Type Definitions dialog and type in the XML text that appears in **Fig. 21**.

The above example creates a complex data type called "Geek" that has two separate sub-components, "Name" and "Salary" of type "string" and "double" respectively. As depicted in **Fig. 23.,** *the new data type "Geek" is available to choose when creating a task or net variable. Variables with a usage of "Local" can have initial values specified for them, as depicted in the same figure. As with the data type definition dialog, parse errors will be listed when the initial value text is red.*

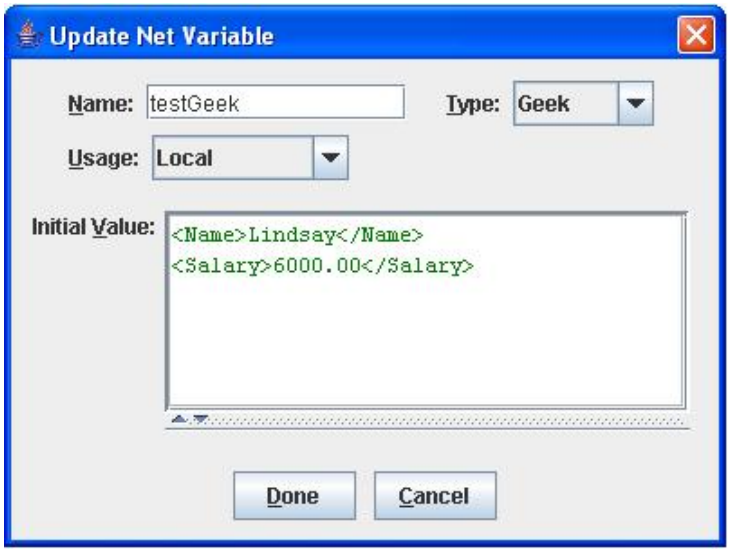


Fig. 23. A "Geek" task variable

Net Decomposition Detail / Updating Net Variables

You can add variables to a net to store information relating to that net that tasks within the net may need to read or update.

To add a variable to a Net:

1. Choose **Update Net Detail** from the **Net** Menu.

We will be setting up Net variables in the “Attend University” net.

Go to the “Attend University Net” and choose **Update Net Detail** from the **Net** Menu.

2. An Update Variables dialog box will appear. **Fig. 24.**
3. Click on **Create**. An Update Net Variable dialog box will appear.

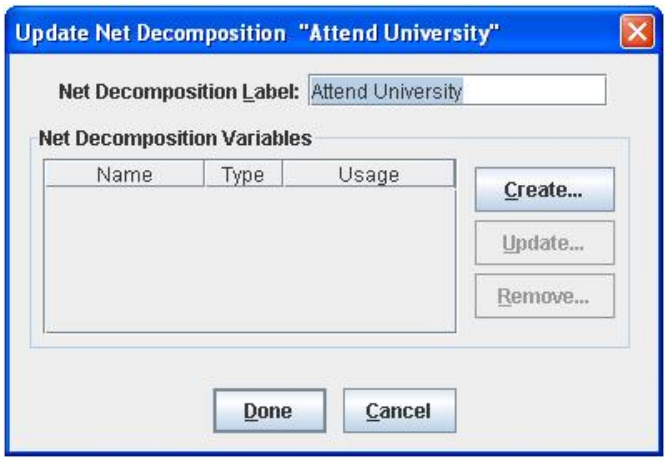


Fig. 24. Updating "Attend University" Net Variables

4. Enter the Name of your variable, choose the Type of the variable from the list, then click **Done**, then **Done** again to close the Net Decomposition dialog.

Type in “StudentNumber” for the name of the variable and choose its type as **string**. Set the Usage to “Local”. **Fig. 25.** Click **Done**.

Create another Net Variable with the name “SubjectCode” and Type of **string**. Leave the Initial Value blank and set the Usage to “Local”. Click **Done**.

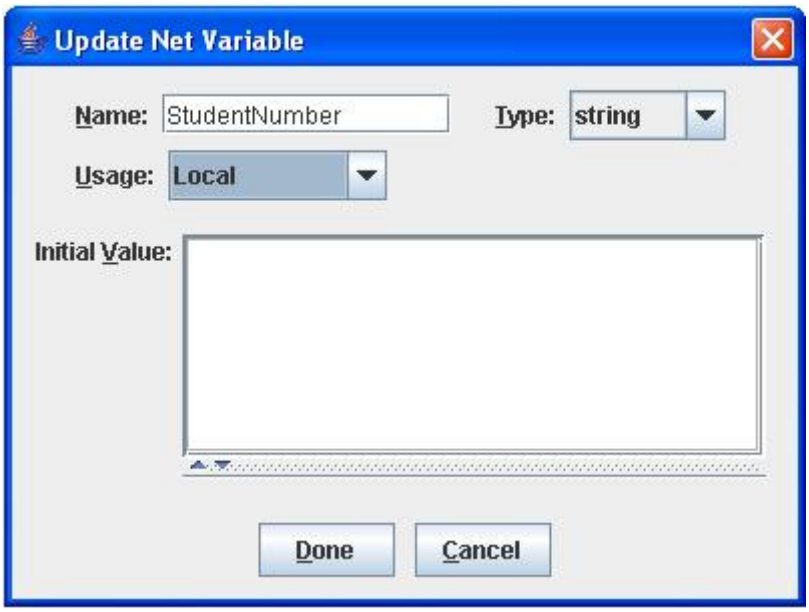


Fig. 25. The Net Variable "StudentNumber"

5. The Net Variables should now appear in the Update variables of Net “Attend University” dialog box. **Fig. 26.**

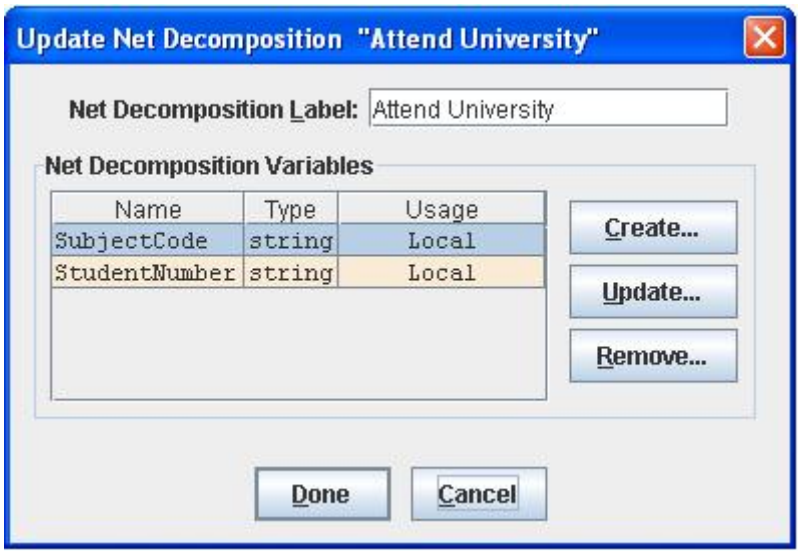


Fig. 26. Updated "Attend University" Net Variables

Select Task Decomposition

By choosing the **Select Task Decomposition…** option when you right click on a task, you have the ability to identify which decomposition this task represents.

Like nets, tasks have decompositions where you can specify variables and a label to associate with the task or net. Unlike nets, which cannot share net decompositions, there is a 1:N relationship between task decompositions (scoped to the entire specification) and their tasks (scoped to nets).

Besides variables and a label, task decompositions also allow the workflow designer to identify which web service the decomposition should invoke in a running workflow engine. When two tasks share the same decomposition, we are saying that the same activity is required in two different places in the workflow.

You can use the drop-down list to select the task decomposition of a task, or alternately you can press the **Create…** button and generate a new one that will automatically become the task's decomposition. **Fig. 27.**

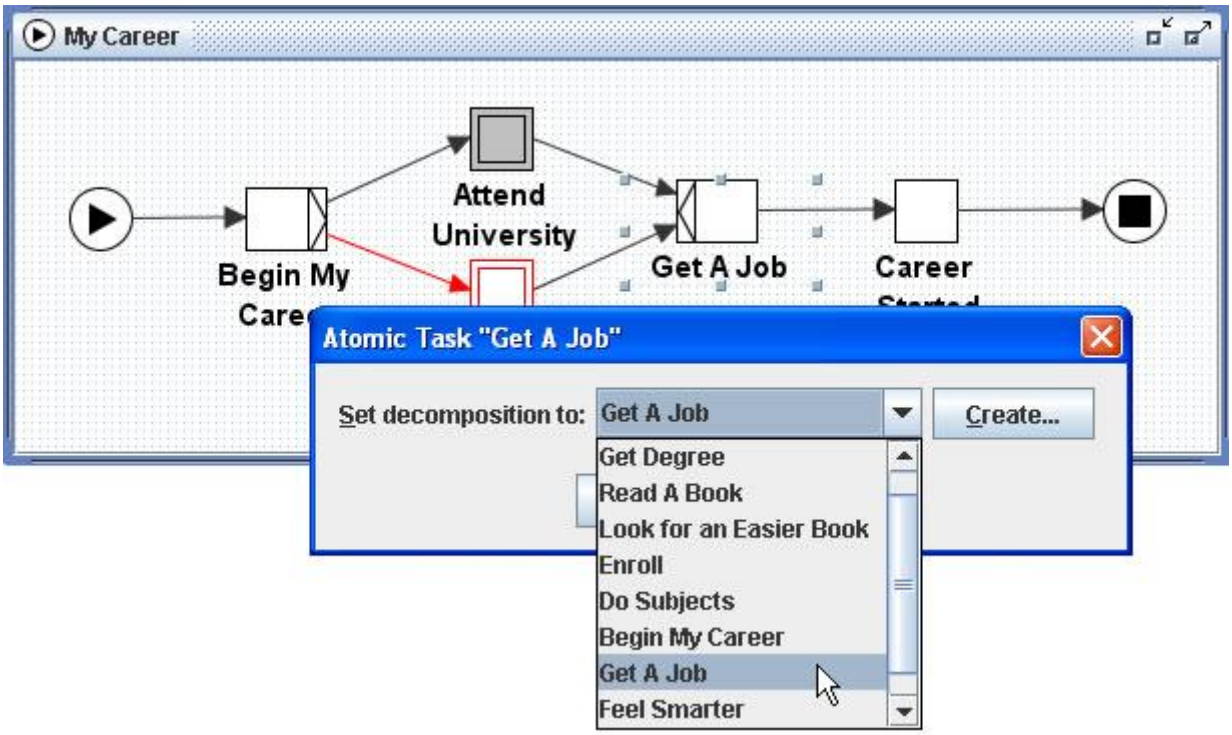


Fig. 27. Select Task Decomposition

Task Decomposition Detail / Updating Task Variables

You can add variables to a task to store specific information relating to that task. Task variables have several uses. One use is to transferring information between workflow users and the workflow engine. A second use is for passing data between web services that the running workflow engine invokes and the Net the task resides in.

For example, if your task is called ‘Place an Order’, you may want to store the name or identification number of the person placing the order or maybe store the items being ordered.

Adding a Variable to a Task

1. First select the task that will require the variable, using the Enter Marquee Selection Mode button, , from the Palette Bar.

We will be setting up variables for the “Enroll” task.

Go to the “Attend University” Net and select the “Enroll” task.

2. Right-click on the task and choose **Task Decomposition Detail…**. An Update Task Decomposition dialog box will appear. **Fig. 24.**

Retrieve the decomposition detail for the “Enroll” task.

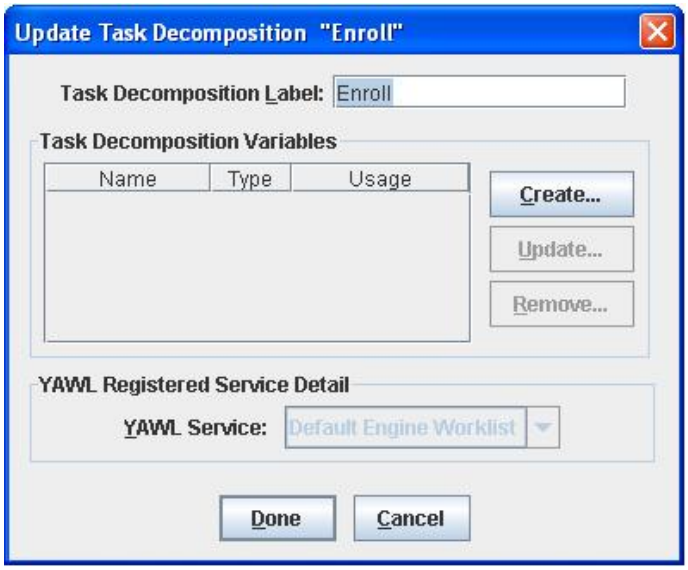


Fig. 28. Updating variables for a task

3. Click on the **Create…** button. An Update Task Variable dialog box will appear.

4. Enter the Name of your variable, choose the Type of the variable from the list, click **Done**, then **Done** again to exit the task decomposition detail dialog.

Type in “StudentNumber” for the name of the variable and choose the type **string**. Leave the Initial Value blank. **Fig. 29.**

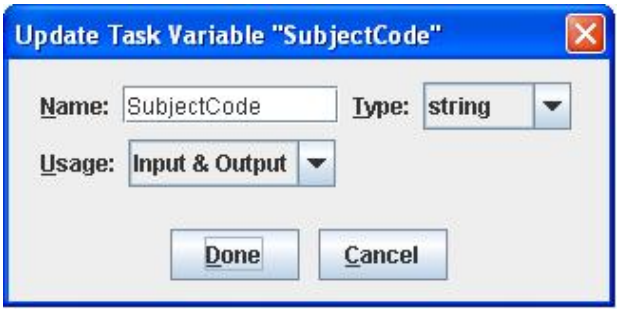


Fig. 29. Updating the Task Variable

5. Create another variable for the same task, called “SubjectCode” with the type being **string** and usage being “**Input & Output**”**.** Click **Done**. **Fig. 30.**

The “Enroll” task now has two variables, “StudentNumber” and “SubjectCode”.

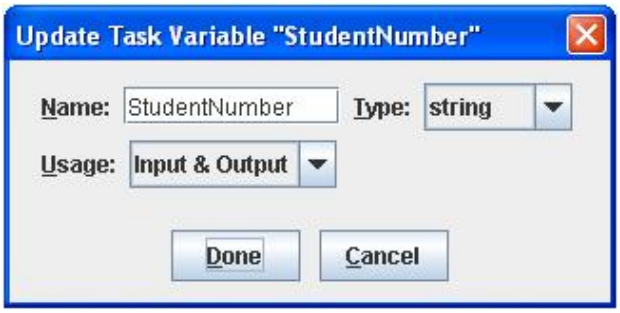


Fig. 30. Updating another Task Variable

Adding / Updating Task Parameters

Both Input and Output Parameters can be assigned to any tasks to allow the passing of state between nets and their tasks, and between tasks and workflow engine users and web services.

Input Parameters use an XQuery to massage net variable state (across possible several net variables) into a value that can be passed to a single selected task variable.

Output parameters use an XQuery to massage task variable state (across possible several task variables) into a value that can be passed to a single selected net variable.

For example, if a task is called ‘Place an Order’, then an Input Parameter could be the name of the person placing the order, whereas the Output Parameter could be the corresponding identification number of that person.

To add an Input Parameter:

1. First select the task that will require the parameter, using the Enter Marquee Selection Mode button, , from the Palette Bar.

   We will be setting up Input Parameters for the variables that we created in the Adding / Updating Task Variables section previously.

   Go to the “Attend University” Net and select the “Enroll” task.

2. Right-click on the task and choose **Update Parameters…**. An Update Parameters dialog box will appear. **Fig. 31.**

   Update the Parameters for the “Enroll” task.

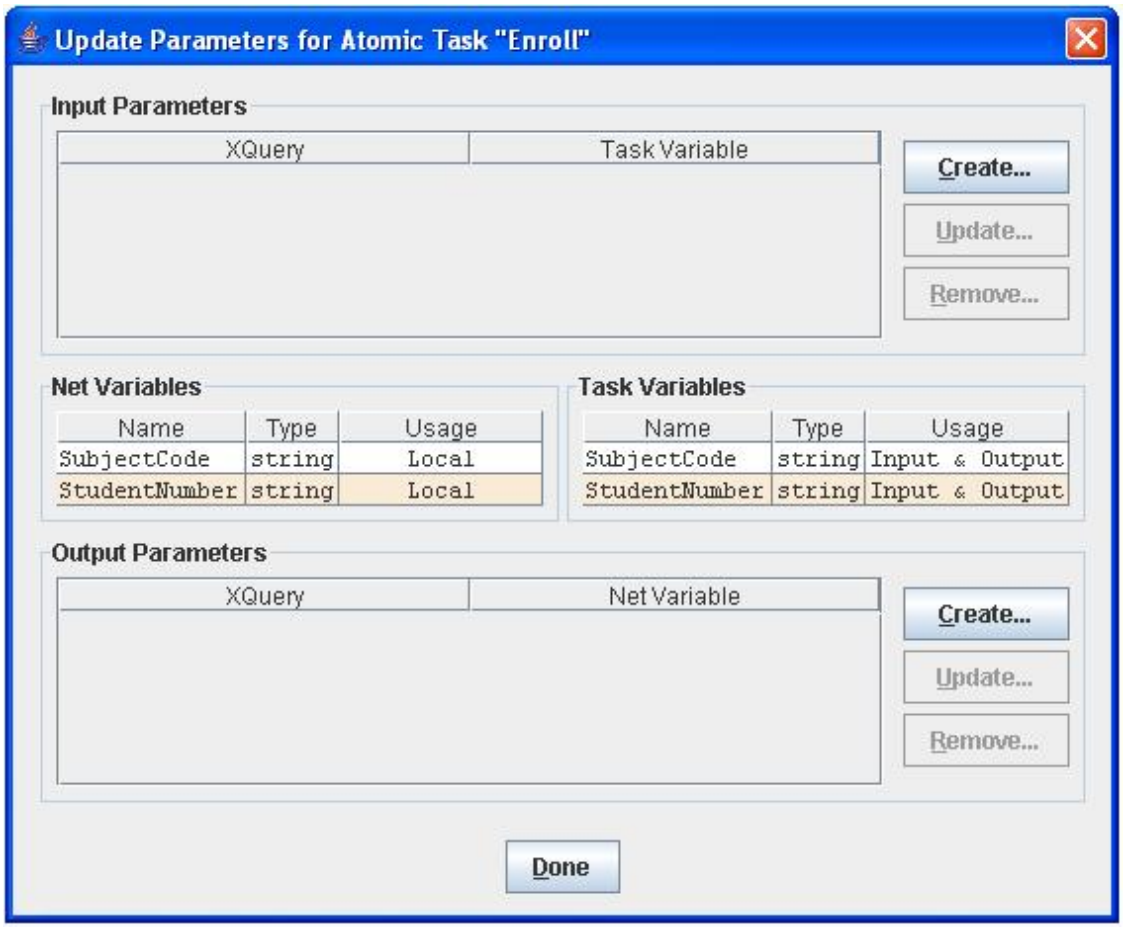


Fig. 31. Updating Parameters for a Task

3. In the Input Parameters section, click on **Create…**. An Update Task Parameter dialog box will appear.

   If you have already set up a Task Variable for this task, then the Existing Task Variable option will be activated and there will be a list of task variables to choose from. Choose a variable from the list and click on **Done**, then **Done** again to close the task parameters dialog.

   If you haven’t set up Task Variables, then click on **Create…** and return to the previous section in the manual for Adding a Task Variable.

   If you are familiar with XQuery syntax, then you can paste in an XQuery

to allow manipulation of the Input Parameters. "Syntactically well formed" XQueries will be green, and badly formed ones will be red. Again, red text will be accompanied by a split-pane table, returning the parse errors that cause the text to be badly formed.

For workflow designer convenience, two XQuery buttons are supplied to generate XQuery expressions from available net variables.

The **add XQuery of element's content only** button will return just the content of the XML element for this variable, which is useful for simple state transfer between two variables of the same XMLSchema type, and expected to be the typical button that users will start out with.

The other button, **add XQuery of entire element**, will return the entire XML element of the selected variable, which is useful for times when you want to create a complex type expression from individual variable elements. Experience with XMSchema and XQuery are necessary to understand the effects this button will have on runtime YAWL engine state.

Select the "SubjectCode" variable from the list of Existing Task Variables. Select the "SubjectCode" net variable and click **add XQuery of element's content only (Fig. 32.**) Click **Done.**

Create another Task Parameter and map the net "StudentNumber" to the task variable of the same name using this technique. Click **Done** and **Done** again.

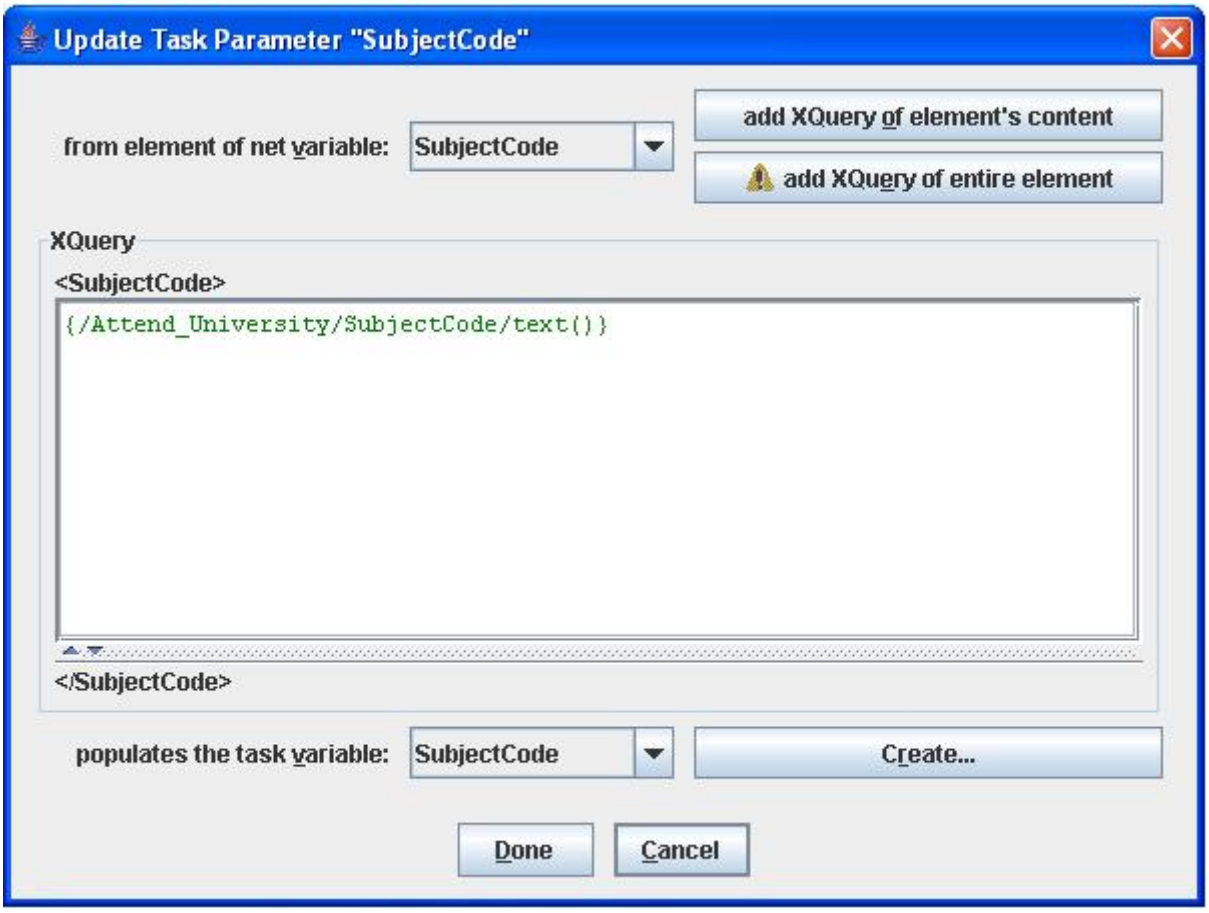


Fig. 32. Passing a net's SubjectCode to a Task

To add an Output Parameter:

1. First select the task that will require the parameter, using Enter Marquee Selection Mode button, , from the Palette Bar.

   We will be setting up Output Parameters for the variables that we created in the Adding / Updating TaskVariables section.

   Go to the "Attend University" Net and select the "Enroll" task.

2. Right-click on the task and choose **Update Parameters…**. An Update Parameters dialog box will appear. **Fig. 31.**

   Update the Parameters for the "Enroll" task.

3. In the Output Parameters section, click on **Create**. An Update Net Parameter dialog box will appear (similar to **Fig. 32.**).

4. If you have already set up a Task Variable for this task, then the Existing Task Variable option will be activated and there will be a list of task variables to choose from. Choose a variable from the list and click on **Done,** then **Done** again to close the task parameters dialog.

   If you haven't set up Net Variables, then click on **Create…** and return to the previous section in the manual for Adding a Net Variable.

If you are familiar with XQuery syntax, then you can paste in an XQuery to allow manipulation of the Output Parameters. "Syntactically well formed" XQueries will be green, and badly formed ones will be red.

For workflow designer convenience, two XQuery buttons are supplied to generate XQuery expressions from available task variables.

The **add XQuery of element's content only** button will return just the content of the XML element for this variable, which is useful for simple state transfer between two variables of the same XMLSchema type, and expected to be the typical button that users will start out with.

The other button, **add XQuery of entire element**, will return the entire XML element of the selected variable, which is useful for times when you want to create a complex type expression from individual variable elements. Experience with XMSchema and XQuery are necessary to understand the effects this button will have on runtime YAWL engine state.

5. From the list of task variables, select the "SubjectCode" task variable and click **add XQuery of element's content only**. From the list of net variables, select the "SubjectCode" variable. Click **Done.**

Create another Task Parameter and map the task "StudentNumber" to the net variable of the same name using this technique. Click **Done** and **Done** again.

These Output Parameters were set up to demonstrate a simple transfer of state from a net to a task and back to the net. Perhaps the task would allow a user to change the values of one of the variables which would eventuate in the net's values changing.

The Update Parameters dialog box should appear as in **Fig. 33.**

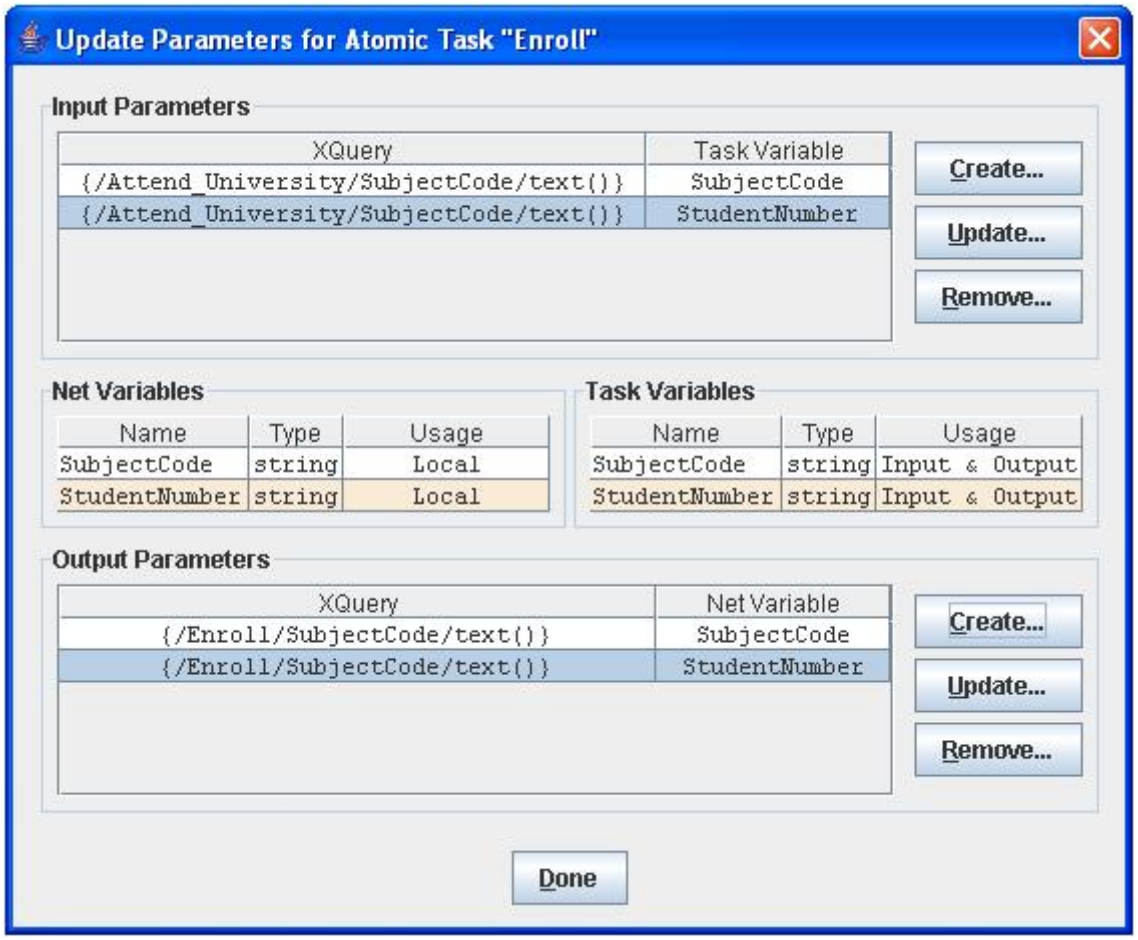


Fig. 33. Established task parameters

A brief introduction to XQuery:

- XQuery is a separate language that describes how to manipulate a given XML document to extract data from it. For example, the XQuery "/data/myTaskVariable/number() + 5" is saying that there is some XMLSchema element in an engine XML document with an element called myTaskVariable whose content is a number, and that the resultant XQuery expression is that element's value with 5 added to it.

Updating Flow Detail

When dealing with tasks that have XOR and OR splits, we need some way of telling a running engine which flows should be activated. By updating the flow detail, you can specify the default flow path and also establish predicates for each flow direction. Flows whose predicates evaluate to true in a running engine will execute those flows.

To update the flow detail of a task that has a split:

1. Right click on the task and choose **Update Flow Detail…**.

   The number of splits will be the number of Target Tasks. The currently

selected flow in the dialog will also be identified by being drawn green in the Net. **Fig. 34.**

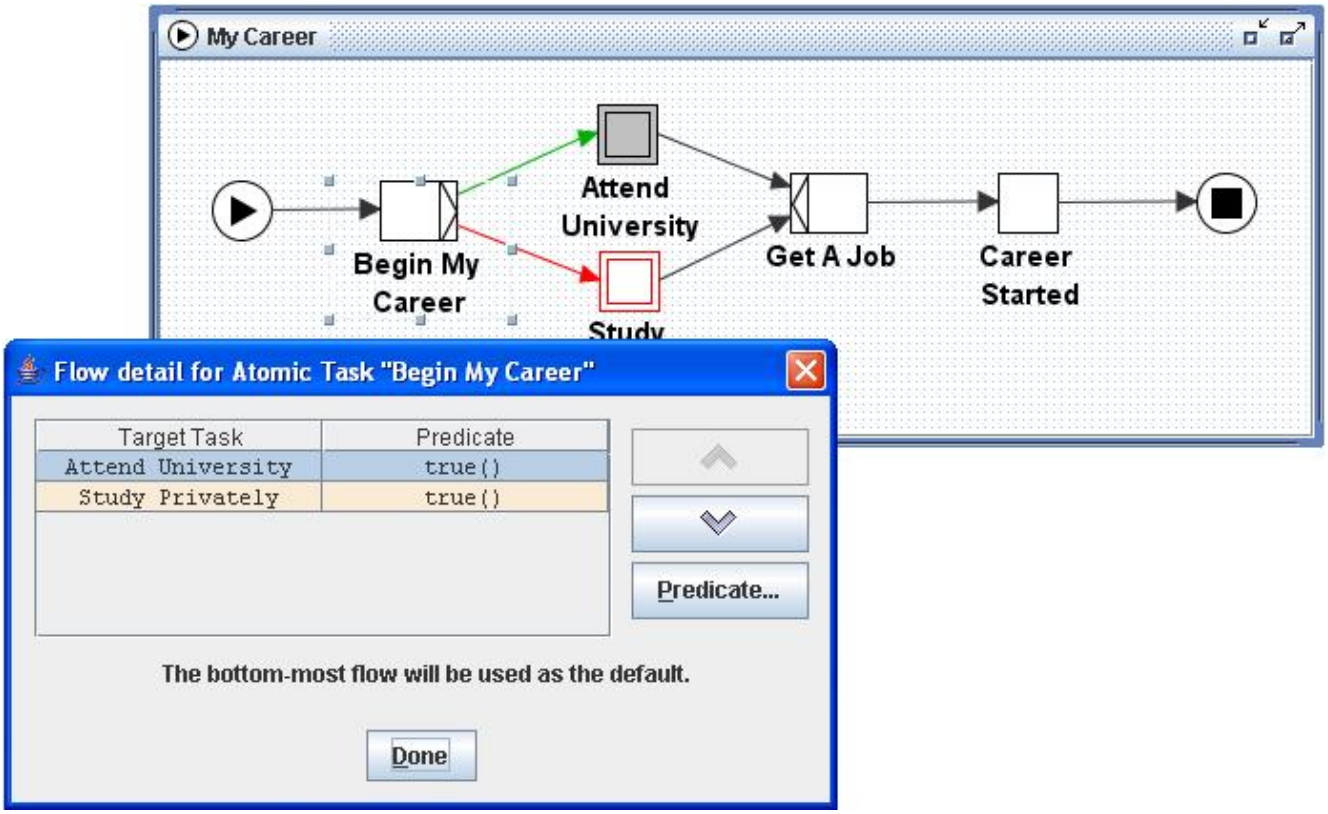


Fig. 34. Updating the Flow Detail

2. To choose the flow ordering, select the Target Task that you want as the change the order of, use the Up and Down arrows on the right to position the flow. The bottom-most flow will be used as the default if some flow needs to be followed, but no predicates evaluate to true.

3. To specify a predicate for a particular flow, select the flow from the list and click on **Predicate…**

   Enter a predicate as a boolean XQuery expression and choose **Done**.

End of Scenario

This is where our scenario ends. The My Career Scenario was designed to explain all the functions of the YAWL Editor and to provide you with a rudimentary understanding of designing a YAWL workflow specification.

But the complexity of the YAWL Editor does not stop with the current scenario. If you are after something more challenging, try adjusting your version of the scenario to expand into more sub-Nets and more complex situations.

The completed example described in this scenario is supplied with the editor source distribution when you're ready to experiment further (**Fig. 35**).

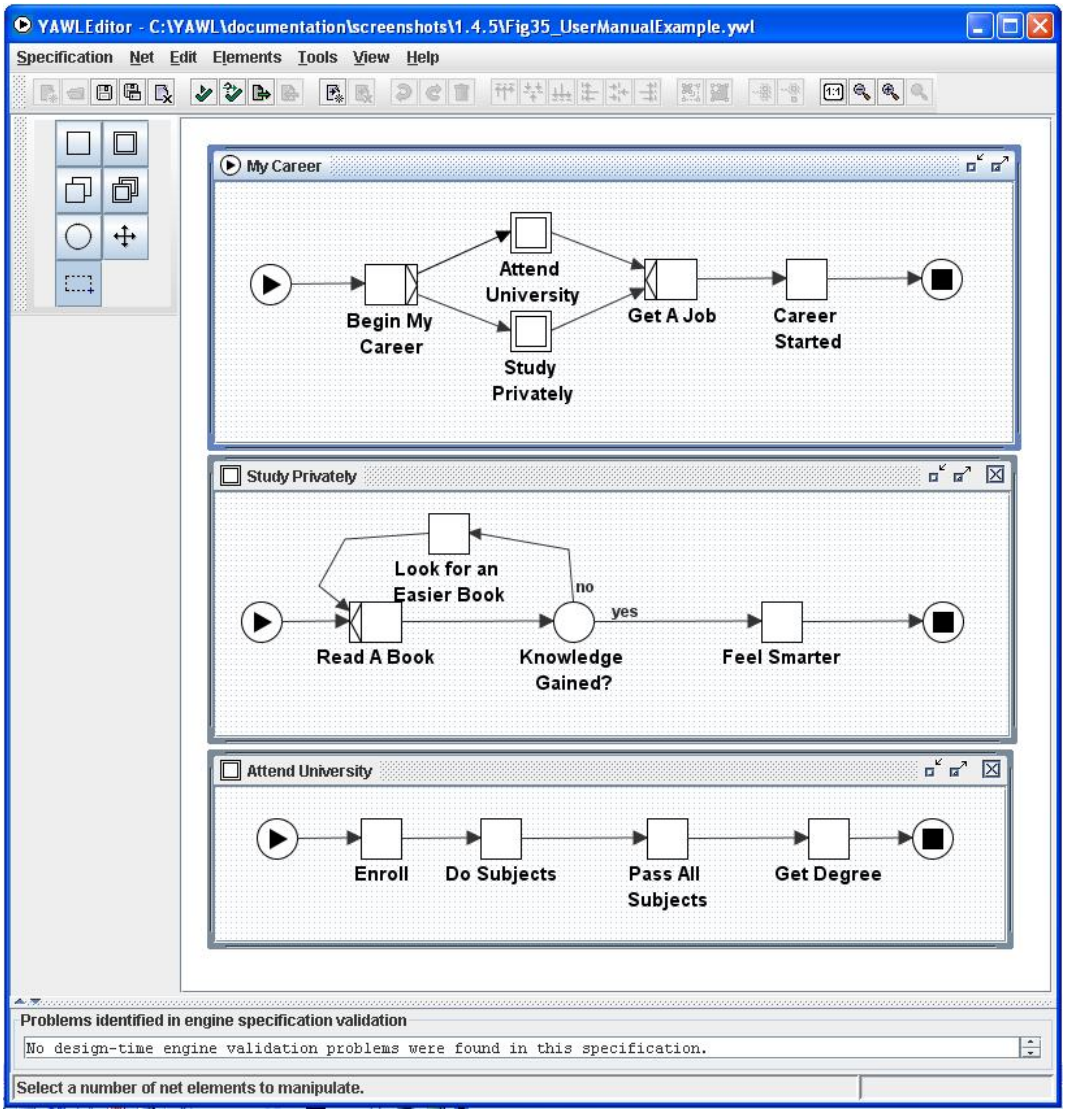


Fig. 35 - The completed scenario specification

Set YAWL Engine Detail

By specifying the location of a running instance of the YAWL Engine, it is possible to have the editor generate specifications that take advantages of that YAWL engine's specific setup.

For this release of the editor, the engine behaviour available is limited to a number of web services that the engine has been configured to interact with. Further detail on specifying web services to have the workflow interact with can be found in the next section.

To set the YAWL Engine details:

1. Click **Engine** on the Menu and choose the **Set Engine Detail…** item**.**
2. From the resulting dialog (**Fig. 36.**), accept the default values or enter the following engine details:
    - YAWL Engine URI
    - User Name
    - Password
    - Verify Password

The YAWL Engine URI value will default to a locally installed YAWL Engine administration URI. The User Name and Passwords will default to the engine default "admin" user.

Fig. 36. Specifying the YAWL Engine Details

As a user convenience, a button called **Test Connection** is supplied, which will attempt to connect to a running engine with the detail supplied before the user commits to using those details for any further editor/engine interactions.

Connecting a Decomposition to a registered YAWL Service

You can use task decompositions within your workflow to make a connection to custom YAWL services that have been registered with a running engine.

For example, a decomposition may be set up to place an order with an external company. Upon execution of any task using this decomposition, data could be transmitted via a Web Service invocation to this company.

To have a decomposition invoke custom YAWL service, do the following:

1. Right-click on a task, the choose **Task Decomposition Detail…**. An Update Task Decomposition dialog box will appear. (see the "YAWL Registered Service Detail" box of **Fig. 28.**).
2. Enter the following details regarding the Web Service:
   - **YAWL Service** – the Service registered with the YAWL Engine
3. Click **Done** to finish.

If a valid running YAWL Engine instance can be connected to via the detail supplied in the section above, the YAWL Service Dropdown Box will contain entries for all custom YAWL services the engine has registered. Otherwise, only the default (manual) "Worklist" YAWL service will be available.

When you select a YAWL Service, the editor will query this running engine for the input and output variables required, and populate the decomposition variables of the selected task with those variables. Base custom services that are supplied with the engine include one for RPC-Style Web Service Invocation, and one for making and receiving mobile phone SMS messages. Later in this manual, an example editor save file, showing the usage of a custom time service is supplied.

Setting Multiple Instance Queries

Multiple Instance Tasks need extra detail for data manipulation. The "Queries" tab, available by right-clicking on a multiple-instance task and selecting **Set Instance Detail…**. The "Queries" tab allows you to manipulate multiple instance data for the task.

Right click on the task and choose. Ensure that you are viewing the "Queries" tab of the dialog, as per **Fig. 37.**

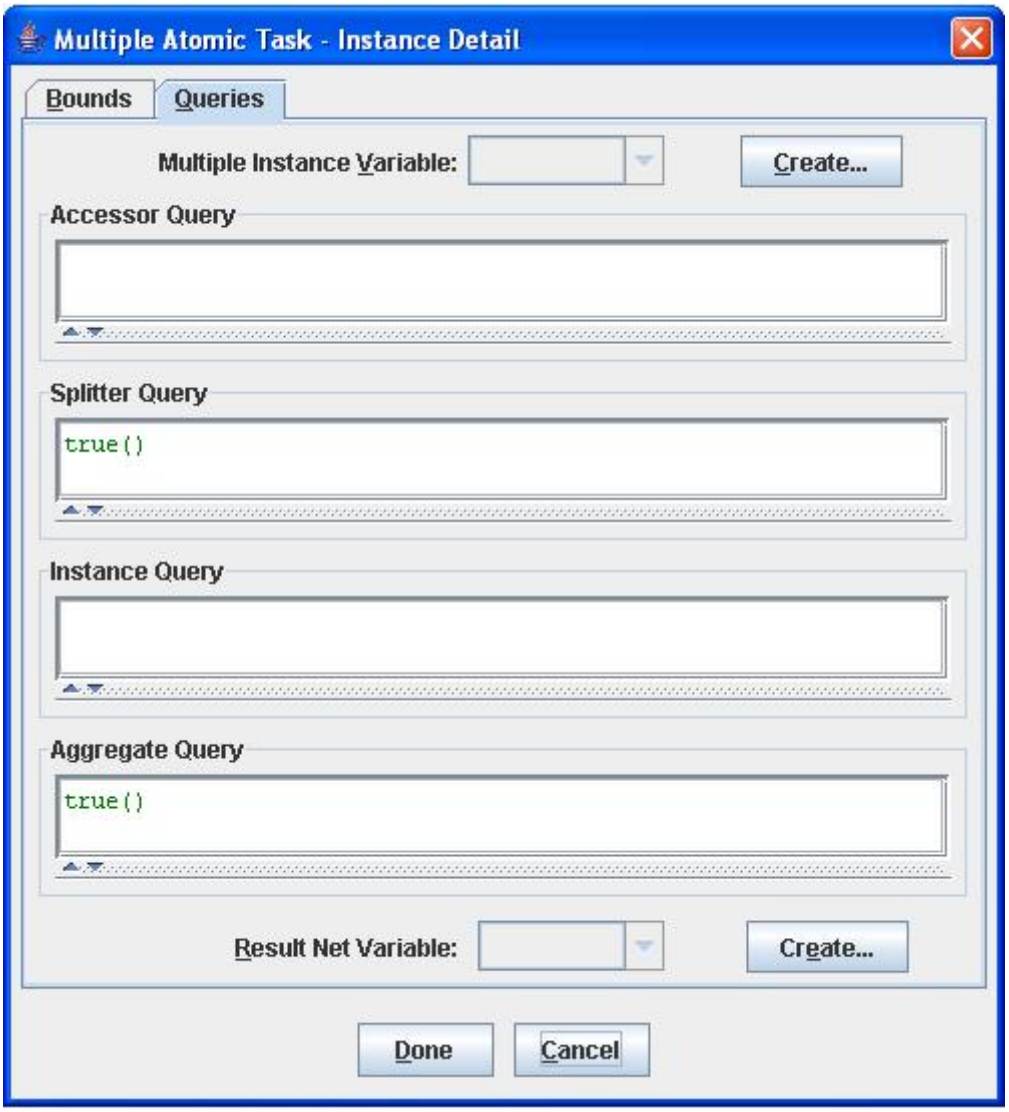


Fig. 37. Data Manipulation for Multiple Instance Tasks

To successfully specify multiple instance data manipulation, you first need to select which of the task's decomposition variables will be used as the "Multiple Instance Variable". This variable needs to contain (or can derive) a number of unique values that will be separated with the Splitter query to pass a unique value per each instance task. If the "Multiple Instance Variable" needs manipulation overall before the unique values are split out of it, use the Accessor Query to do so.

On the completion of an instance, the "Instance Query" can be used to transform the XML document returned to a form suitable for the Aggregate query to finally generate an overall result. The overall result will be assigned to the "Result Net Variable" on completion of the multiple instance task.

See **Fig.** for an example of this screen being filled out with valid XQuery expressions.

Validating, Exporting and Importing

At any stage you can validate or export your specification to a YAWL Engine executable XML file.

To Validate your specification:

1. Click on Validate this Specification button, , on the Menu Toolbar or click **Specification** on the Menu and choose **Validate Specification…**.

2. A table listing problems will appear with details of any inconsistencies that would stop the specification from running in the YAWL Engine in the specification problem panel. **Fig. 38.** shows an example valid specification. **Fig. 39.** shows an example invalid specification.

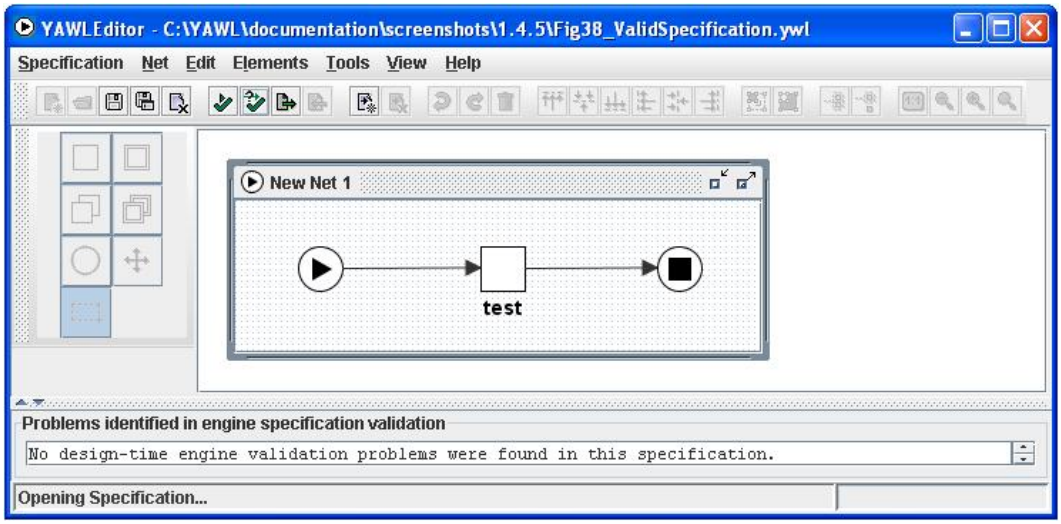

YAWLEditor - C:\YAWL\documentation\screenshots\1.4.5\Fig38_ValidSpecification.ywl
Specification Net Edit Elements Tools View Help
New Net 1
test
Problems identified in engine specification validation
No design-time engine validation problems were found in this specification.
Opening Specification...

Fig. 38. A valid specification

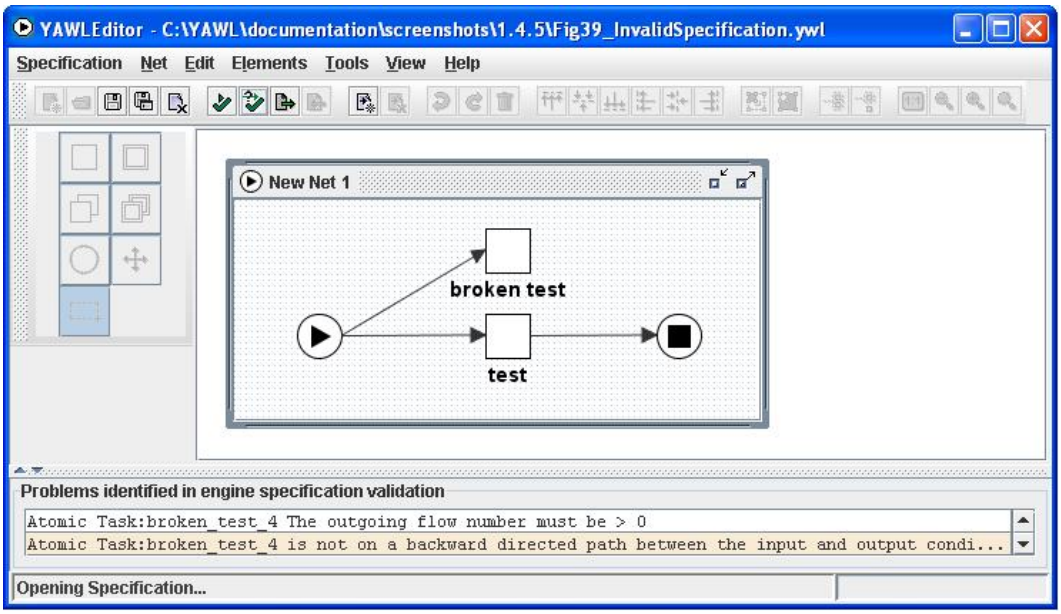

YAWLEditor - C:\YAWL\documentation\screenshots\1.4.5\Fig39_InvalidSpecification.ywl
Specification Net Edit Elements Tools View Help
New Net 1
broken test
test
Problems identified in engine specification validation
Atomic Task:broken_test_4 The outgoing flow number must be > 0
Atomic Task:broken_test_4 is not on a backward directed path between the input and output condi...
Opening Specification...

Fig. 39. An invalid specification

To Export your specification to an XML file:

1. Click on the "Export this specification to the YAWL Engine file format" button, , on the Menu Toolbar or click **Specification** on the Menu and choose **Export to YAWL Engine File…**.
2. A window will appear asking you where to save the file.
3. Select a location and choose **Save**.

This specification save file can now be loaded into a running YAWL Engine and executed.

New for this release of the editor is the ability to import from an engine XML file:

4. Click on the "Import YAWL Engine file" button, , on the Menu Toolbar or click **Specification** on the Menu and choose **Import from YAWL Engine File…**.
5. A window will appear asking you specify the engine XML file.
6. Select a location and choose **Open**.

This engine XML file is then loaded into the editor. As there is no layout and graphing detail on the engine XML file, the editor will apply basic layout to the imported specification.

Specification Analysis

Verification of specifications for the engine only determines whether the specification will be executed by the engine. In contrast, the analysis tool can be used to test for deeper issues in the specification.

The analysis toolbar button, , and matching **Analyse Specification** menu item under the **Specification** menu allows workflow designers to analyse their specifications. A number of potential problems with the workflow can be automatically spotted with analysis. Examples include spotting potential deadlock situations, unnecessary cancellations set members, and unnecessary or-join decorators (at run-time, or-joins require significant processing effort, and should be removed if they are not actually needed).

A configuration dialog for the specification analysis is available under the **Tools** menu, by the name of **Configure Analysis Tool** (see **Fig. 40**).

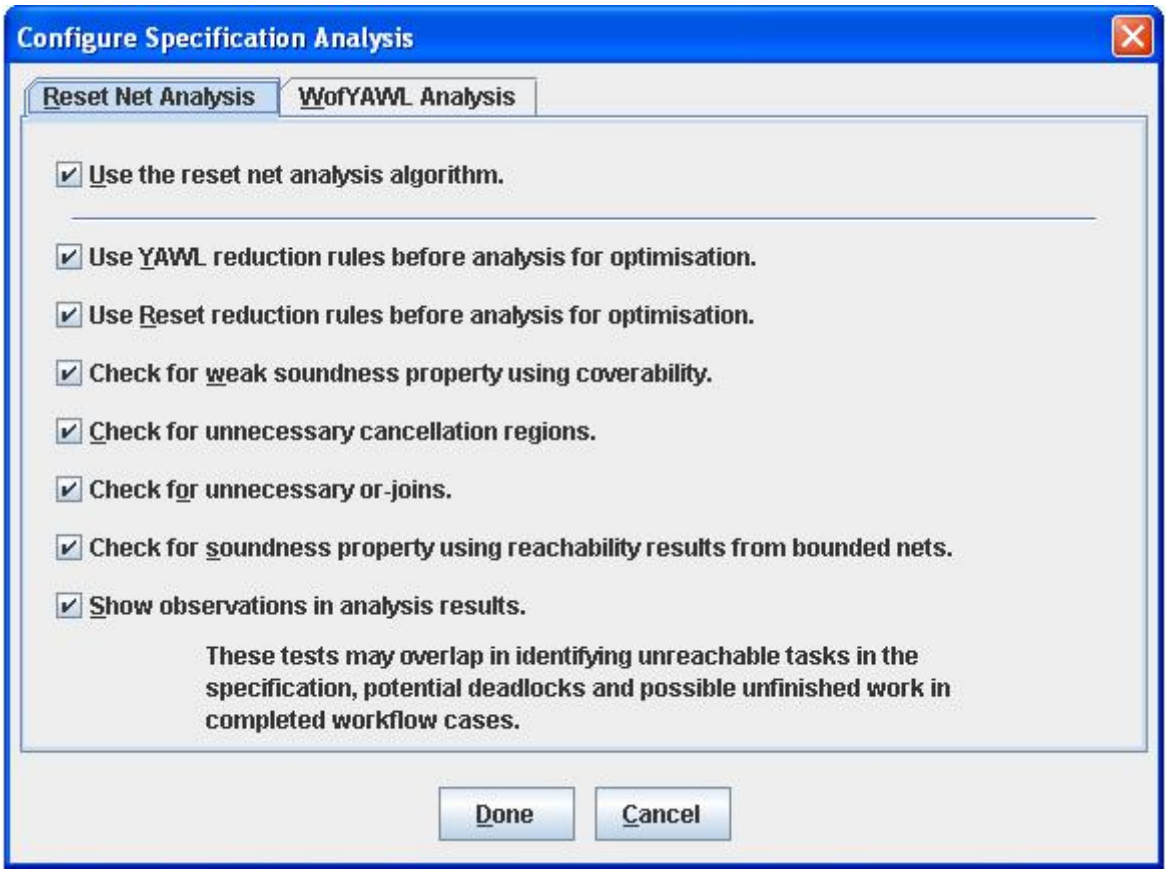

Configure Specification Analysis
Reset Net Analysis
WofYAWL Analysis
Use the reset net analysis algorithm.
Use YAWL reduction rules before analysis for optimisation.
Use Reset reduction rules before analysis for optimisation.
Check for weak soundness property using coverability.
Check for unnecessary cancellation regions.
Check for unnecessary or-joins.
Check for soundness property using reachability results from bounded nets.
Show observations in analysis results.
These tests may overlap in identifying unreachable tasks in the specification, potential deadlocks and possible unfinished work in completed workflow cases.
Done
Cancel

Fig. 40. Specification analysis functions

If the optional YAWL specification analysis utility **wofyawl.exe**, written by Eric Verbeek is supplied in the same directory as the editor, an extra tab entitled **WofYAWL Analysis** will be enabled in this dialog, allowing more analysis options than those supplied by default. The utility must be compiled for specific architectures, and its data format has changed over time. This version of the editor needs version 0.4 of the utility, and expects it to be called **wofyawl0.4.exe**, regardless of the platform on which it runs. Choose the wofyawl implementation right for your environment from Sourceforge, and then rename as above it to get analysis working.

The default specification analysis algorithm was donated to the editor by Moe Wyn, and is part of her research into using Reset Net analysis techniques within the setting of YAWL.

Illustrative Examples

Multiple Instance Task Example

The example save file below, supplied with the editor source, gives you an example of how to process a simple multiple instance task, with its queries filled in (**Fig. 41**).

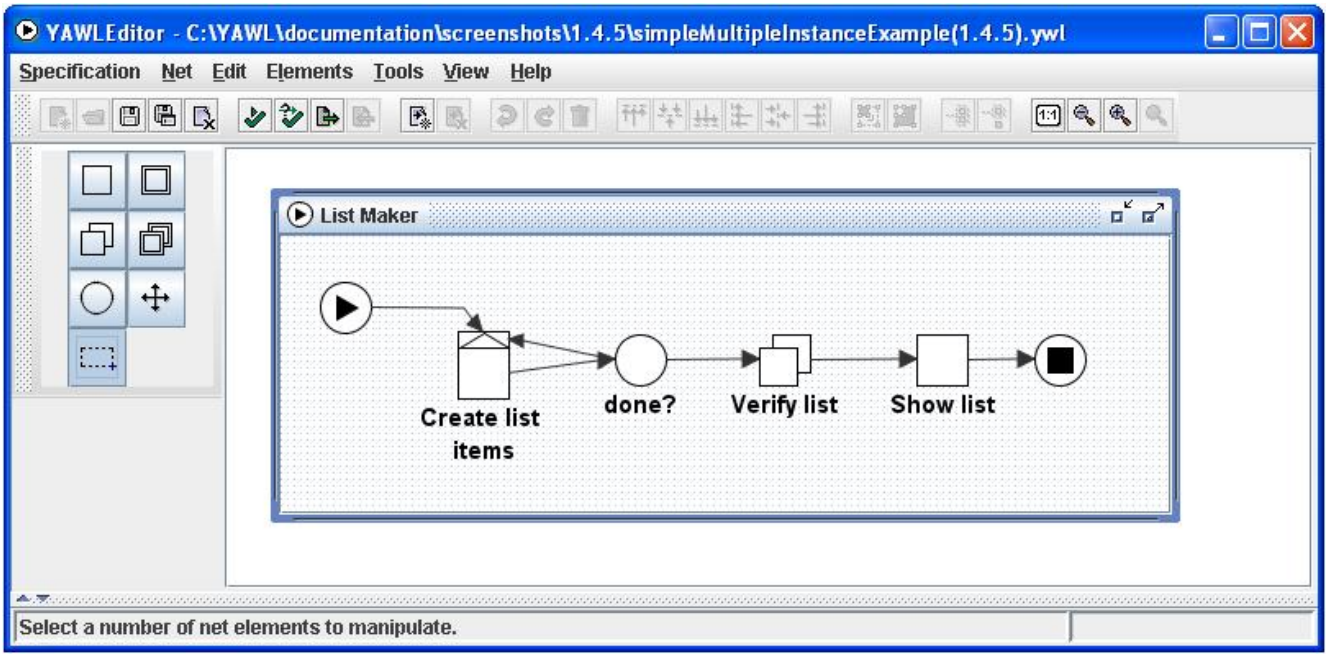


Fig. 42- A simple specification showing how multiple instance tasks are done.

Fig. 43 shows the core multiple instance queries used to manipulate a XMLSchema element defined for this specification. The element has a complex type of “PersonList” which defines a sequence of “<PersonName>someName</PersonName>” elements.

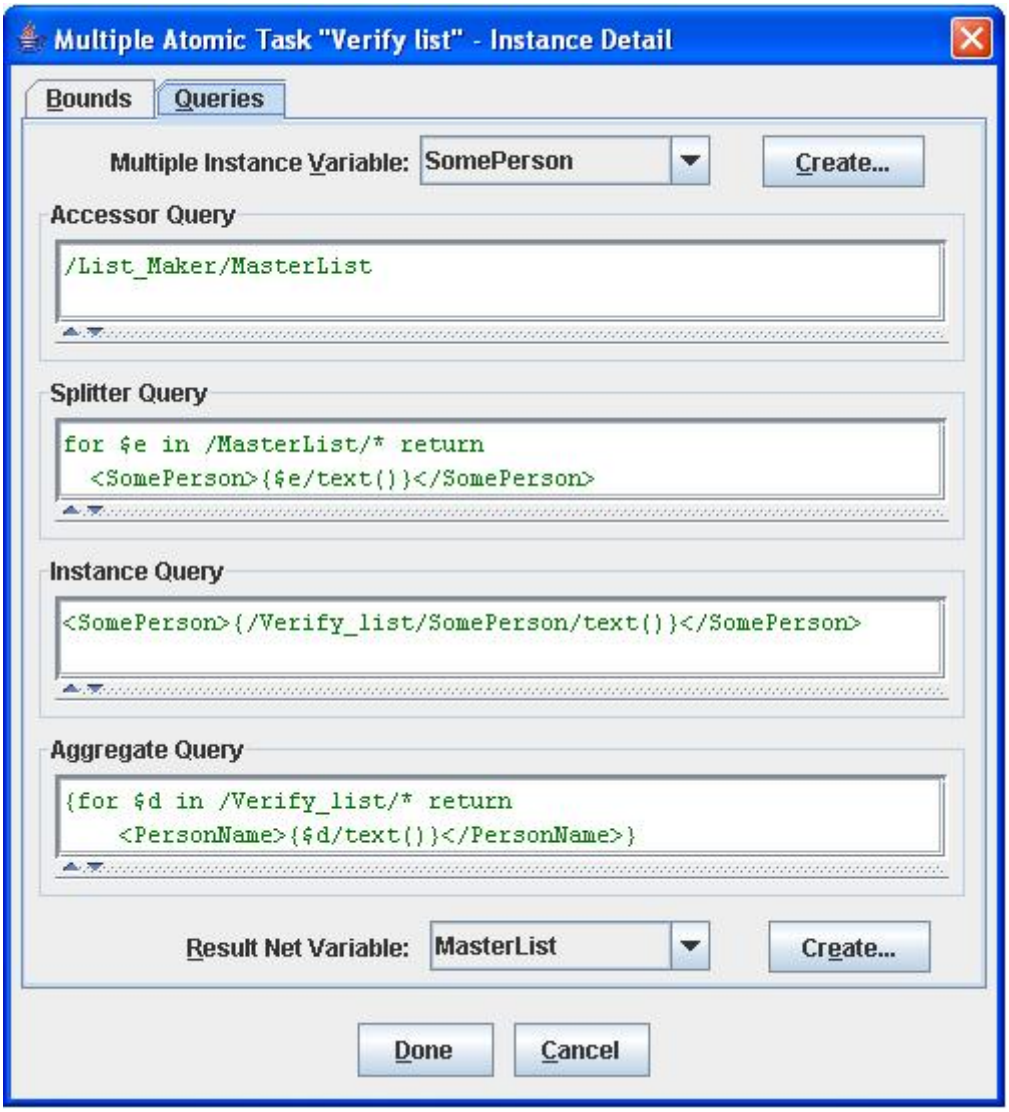


Fig. 44 – Example multiple instance queries

Time Service Example

The example save file below, supplied with the editor source, gives you an example of how to build a task that calls through to the optional timer service supplied with the engine (**Fig. 45.**).

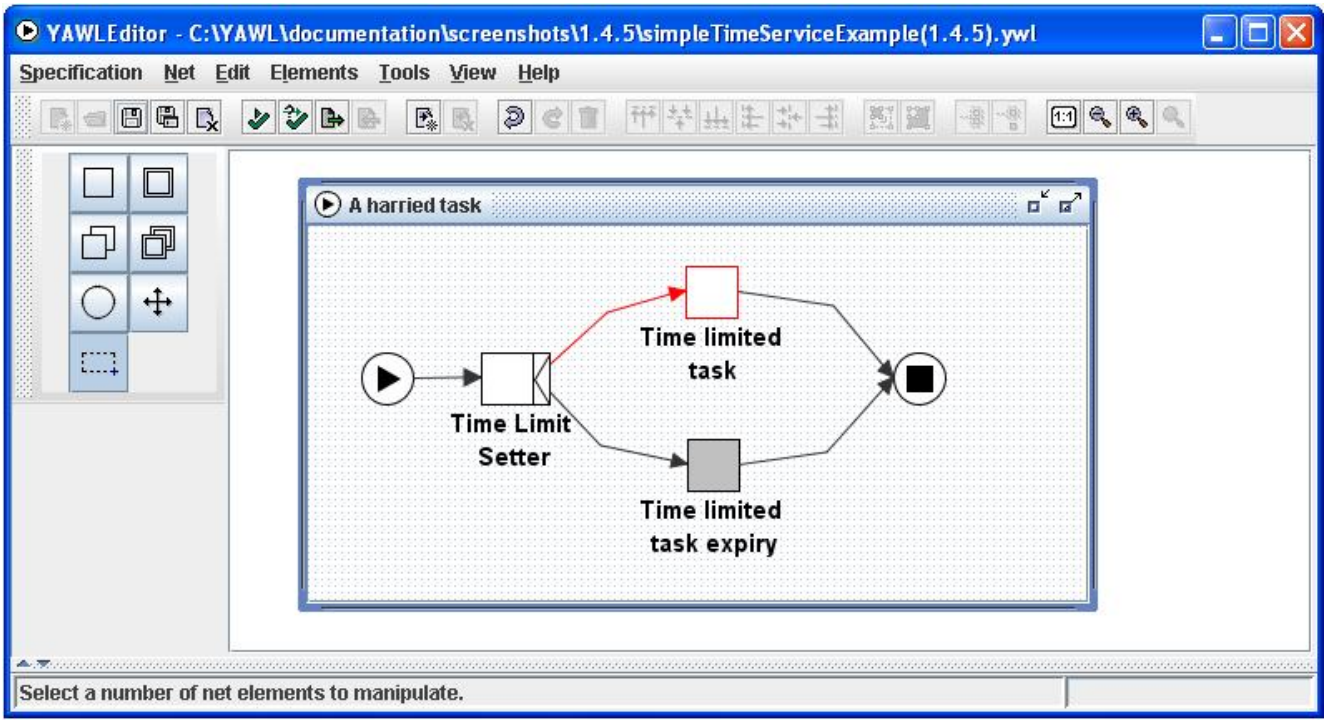

YAWLEditor - C:\YAWL\documentation\screenshots\1.4.5\simpleTimeServiceExample(1.4.5).ywl
Specification Net Edit Elements Tools View Help
A harried task
Time limited task
Time Limit Setter
Time limited task expiry
Select a number of net elements to manipulate.

Fig. 45 - Example showing the usage of a custom YAWL service

Note the inclusion of the flow between tasks “Time limit setter” and “Time limited task”. Without including this flow in the cancellation set of ‘’Time limited task expiry”, even though the “Time limited Task” is cancelled, there could still a token on the flow going to “Time limited task”. Users may suddenly find that there is a time limited task available after the expiry time that triggers cancellation.

Fig. 46 shows a user selecting the time service currently registered with a running YAWL engine as they generate their specification. One of the differences between this release of the editor and previous releases is that custom services registered with a running engine will tell the editor what input and output variables that service is expecting. The editor supplies these variables to workflow designers automatically in this release.

Note that the variables defined by custom services are the “minimum” required. Extra, user-defined variables can still be defined, and some custom service may make use of these option variables.

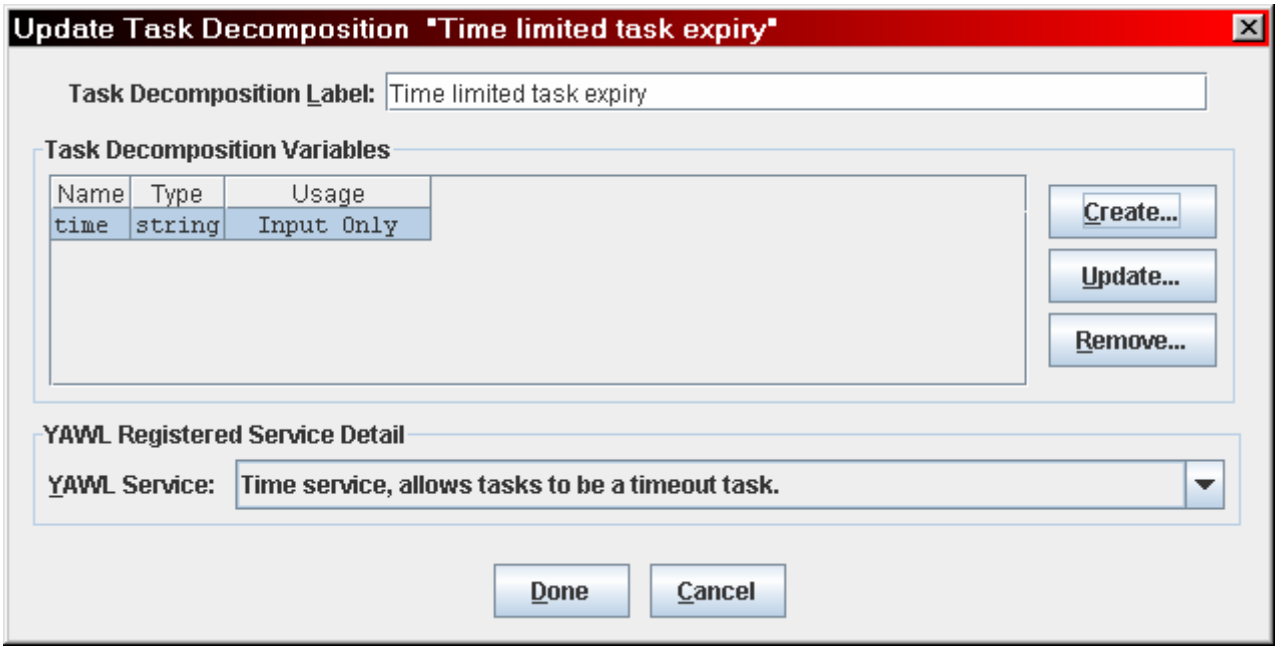


Fig. 46 - Assigning a registered YAWL service from a running engine

Resource Management Example

The example save file below, supplied with the editor source, gives you an example of how to build a tasks that specifies resourcing constraints, such which person or role is authorised and/or allocated to process a task. (**Fig. 47.**). Resourcing constraints can be specified from a new popup menu item available for atomic tasks called **Manage Resourcing**.

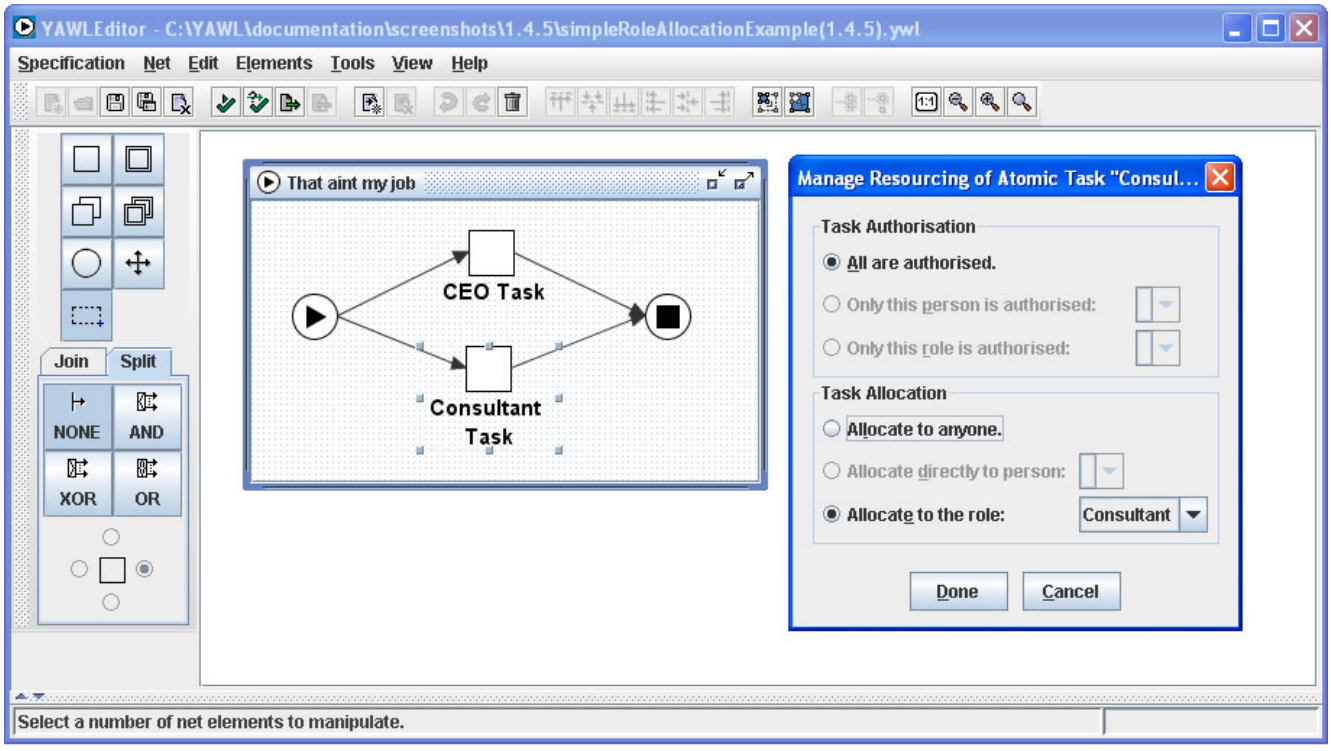


Fig. 47 Managing Resourcing of a task

The dialog for managing resources depicted in **Fig. 47** shows an example of what happens when the editor fails to gain access to an organisation database, but remembers what was last supplied to the editor when the specification was being designed. The workflow designer can still choose between the default and the option where a value was remembered only.

When the editor successfully connects to the organisation database, the drop-down boxes for allocation and authorisation will be populated out of the database and will allow the workflow designer to select from the roles and individuals listed.

When workflow specifications are exported to the engine, SQL queries that identify users or roles authorised for, or to allocated tasks will be automatically generated from the editor and passed into the engine XML. Then engine can then use these queries to shortlist and pick from valid candidates.

More detail on engine run-time user allocation and authorisation can be found in the YAWL Engine manual.

More detail on populating the organisation database of the engine can be found in the YAWL administration tool manual.

Known Issues

Users cannot type double quote characters in XQuery and XPath expressions (this is a deliberate restriction to stop oddities in engine export manifesting). Workflow designers are to use single quote characters when manipulating string literals in XQuery/XPath expressions.

There are intermittent reports of editor save-file corruption occurring with versions of the editor up to and including version 1.2. The 1.3 release of the editor fixes some bugs that could lead to save-file corruption. If you get version 1.3 (or later) editor save-file corruption, we'd appreciate you supplying as much detail to us as possible on when/where/how it occurred to help us nail this problem once and for all. Obviously, working on backup copies of a workflow specification is suggested to avoid any wailing and gnashing of teeth in the meantime.

If you suspect editor save file corruption, run your editor from the command line, and be on the lookout for exception stack traces. The detail of these stack traces is needed for tracking further problems of this nature.

Only the initial values defined for net "Local" variables will influence running engine workflows. All other usage types ("Input Only", "Output Only" and "Input & Output") will not make use of initial values defined for them in the running engine.

The editor does not currently support XMLSchema elements when defining data to pass through a workflow. Hand-created engine specification files that use <element> tags for data transfer will fail on editor import attempts. Convert these into their equivalent <name><type> tags to get editor import working as expected.

Editor Import functionality does not currently import resource perspective detail from an engine XML specification file.

We need to find a way of getting the clipboard to notify listeners of updates. Currently, we kludge up whether the "Paste" action can be activated by simply turning it on when an editor user uses "Cut" or "Copy" in the same editor. Editor elements cannot currently be cut/paste across two running editors on the same machine.

The menu item "Print Specification" freezes if not used for some time.

Troubleshooting

The YAWL Editor is not responding to my mouse clicks and it is beeping every time I click the mouse.

Check to see if you have any YAWL Editor dialog windows open. These windows don't appear in the Windows Taskbar. To check if you have any open dialog windows, hold down the ALT key and press the TAB key. With the ALT key held down, press TAB until you reach the dialog window and let go of the ALT key. Close the dialog window and you should be able use the YAWL Editor again.

Dialog windows appear as the Java icon (blue coffee cup).

I can't connect two elements with a flow. Why?

If the editor does not allow you to connect one element to another, it is steering you away from building an invalid net. Typical examples include:

- Trying to connect a second flow to an undecorated task.
- Trying to point an incoming flow to a split decorator (or an outgoing flow from a join decorator).

Se the troubleshooting entry "How do I find out more about elements and principles....." for more detail.

When I validate my Net, I get the following validation message, 'The net (. . .) may complete without any generated work. Should all atomic tasks in the net be unlabelled?'

This message appears if you have not labelled any of your tasks. To label a task (and thus ensure that the engine will get a user to handle the task at this point in the workflow), right click on the task and choose **Select Task Decomposition…**. Choose **Create…** and in the following window, "Update Task Decomposition", and enter the name of your task under Decomposition Label, then click **Done**.

To label a Condition, right-click on a task and choose **Set Label…**, then type in a name for the label and click **Done**.

My specification won't validate without any errors and I don't know what to do.

First, check the logic of your specification and exercise every branch of your workflow for correctness.

If you are sure of your workflow, consult the YAWL website for the proper use of the YAWL elements:

http://www.yawl-system.com .

How do I find out more about the elements and principles of the YAWL workflow specifications?

For more information about the mechanics of the YAWL workflow elements, please consult the YAWL website:

http://www.yawl-system.com .

Acknowledgements

Queensland University of Technology

Dr Marlon Dumas

Dr Lindsay Bradford

Lachlan Aldred